Secure Mobile Ad-hoc Routing in Maritime Environments

MIKE ALMELOO, University of Twente, The Netherlands

The need for secure and reliable ship-to-ship communication in maritime environments is growing. Because maintaining a centralized infrastructure in this environment is difficult, experimentation is being done on a mobile ad-hoc network called MaritimeManet. Currently, the routing protocol that will be used in MaritimeManet is yet to be determined. MaritimeManet uses multiple directional antennas for each node, which may influence the security of the network in unexpected ways and require the use of a specific routing protocol. We investigate how wireless and routing security risks are affected under MaritimeManet's unique setup, and which consequences this may have for the choice of a routing protocol. Based on these consequences, security requirements for a routing protocol are drafted, and a protocol is chosen that best satisfies these security and practical concerns. This protocol is then implemented in an experimental setup in order to evaluate its use in MaritimeManet.

Additional Key Words and Phrases: MaritimeManet, Ad-hoc Networks, Computer Networking, Routing Protocols, Security

1 INTRODUCTION

With the increase of organizations' operations in maritime environments, a need has arisen for reliable communication between ships at sea. The nature of at-sea operations makes deployment of a centralized network infrastructure very difficult, which calls for a mobile ad-hoc network (MANET). Such a network is currently being developed under the name of MaritimeManet.

MaritimeManet is a mobile, ad-hoc, long-range, broadband network designed for communication between platforms in maritime environments. Examples of such platforms include ships, but also buoys, platforms, helicopters, and Unmanned Aerial Vehicles (UAVs). For wireless communication, a MaritimeManet node uses multiple sectors, each containing two directional radio antennas. These sectors are arranged in a sunflower pattern to cover the complete azimuth. This is in stark contrast to a regular MANET, which typically only uses a single omnidirectional antenna for each node. The fundamental principle in MaritimeManet is to automatically discover other nodes and to set up the strongest possible wireless connection between two nodes within radio reach, by selecting the 'best' sector at both nodes. This process is periodically executed by each node to maintain connectivity while nodes move with respect to each other, which may result in handover of a connection to an adjacent antenna of a node. Movement of nodes may also lead to removal of connections or creation of new connections. The result of all connections between nodes is a mesh network, such as the one shown in Figure 1.

In order to support communication between non-neighbouring nodes in the network, some nodes will need to forward messages to and from other nodes. This is facilitated by a routing protocol, which runs on top of the wireless links that are established by MaritimeManet. Many different routing protocols currently exist, but no definitive decision has been made on which protocol to use. More specifically, MaritimeManet's unique setup with directional antennas may have a positive or negative impact on the security of the network, and research is needed to determine how the choice of a routing protocol affects this decision. Hence, the aim of this paper is to investigate the following questions:

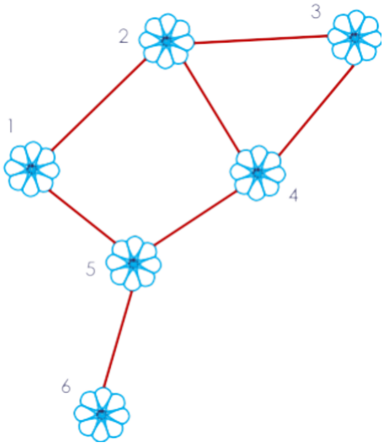

Fig. 1. A MaritimeManet network.

(1) What measures should be taken in order to protect against the most important risks introduced by potential attack paths in the wireless or routing layers of MaritimeManet?
  (a) Which attack paths may be taken by an adversary to compromise the confidentiality, integrity, or availability of MaritimeManet through the wireless or routing layer?
  (b) Which attack paths are most important to address in the context of MaritimeManet?
  (c) Which security requirements should be drafted to mitigate the risks associated with the relevant attack paths?
(2) How does the choice of routing protocol influence the security of a network that uses MaritimeManet?
  (a) Which existing MANET routing protocols provide the best coverage of the drafted security requirements?
  (b) How could a routing protocol be implemented and tested in MaritimeManet?

To answer these questions, we will first look into some of the most important attack paths that an adversary may take to compromise a MANET through the wireless or routing layer. By keeping the unique properties of MaritimeManet in mind, it is possible to create a risk analysis of these attack paths, both for a typical MANET and for MaritimeManet. By analysing the influence that our unique network has on the likelihood and impact of these attack paths, we are able to make an informed decision about the security requirements that a routing protocol should implement in order to keep the network secure. Based on these security requirements and a few practical concerns, we will then compare various routing protocols in order

TScIT 41, July 5, 2024, Enschede, The Netherlands

to find the one(s) most suitable for use in MaritimeManet, and see how one of them could be implemented in an experimental setup.

2 TECHNICAL DETAILS

As mentioned in Section 1, each MaritimeManet node has two antennas per sector. One of these antennas is used for network control, while the other is used for actual data traffic.

The process running at each node that controls the state of wireless connections is called Distributed Neighbourhood Discovery (DND). To provide the information for DND to operate on, the control radio of each sector periodically transmits a 'Sense' message on a predefined, common frequency, and continuously listens for the 'Sense' messages of neighbouring nodes. When DND decides that a connection should be made, it is established through the data radio of the sector. While the control radio currently uses a predefined frequency, frequency hopping is allowed on the data radio by coordination through DND. Work is currently being done on allowing frequency hopping on the control radio as well.

Both the control and data radios use 802.11s, also known as the wireless mesh standard, to communicate with other nodes. One of the key features of 802.11s is Simultaneous Authentication of Equals (SAE): it allows both nodes to initiate the connection and prove to each other that they have a pre-shared key (PSK), without actually revealing it [18]. This process is called peering. SAE only requires each node to send two messages: first, a 'Commit' message is sent to initiate peering, followed by a 'Confirm' message to complete the process. This process also forms the basis for encryption of the wireless link [6, §12.6.1.3.4, §14.5.1].

In order for the nodes to successfully 'find' each other through their data radios, they should both use the same Mesh Basic Service Set (MBSS) identifier. This identifier is roughly equivalent to a network name and is determined by DND. When two adjacent sectors on a node are doing a handover, both sectors are configured with the same MBSS identifier. This allows the handover to happen transparently to higher-layer traffic.

3 EXISTING WORK

There has already been a lot of research on vulnerabilities in mobile ad-hoc networks. Wireless attacks that MANETs are particularly vulnerable to, such as jamming and wormhole attacks, have been discussed in detail. The same goes for various routing attacks [16, 23, 25, 29].

The main shortcoming of existing research when it comes to attack paths is their applicability to MaritimeManet. The research focuses on the general case of mobile ad-hoc networks with omnidirectional antennas, but it is currently unknown what effect MaritimeManet has on these attack paths. Its directional antennas might increase or decrease the likelihood or impact of certain attacks.

Many routing protocols such as Babel [8], OLSR(v2) [9, 10], Castor [14], ARAN [30] and more can be found in literature. A few of these protocols promise mitigation of various security risks commonly present in MANETs. But some of these security measures might be less important in MaritimeManet, while others could be more critical. This paper aims to bridge this knowledge gap by connecting the unique case of MaritimeManet to the existing literature.

4 ATTACK PATHS

In order to determine the threats that MaritimeManet should protect against, we shall first identify the possible attack paths that an adversary could take and draft up a risk analysis. This should include threats in both the wireless and routing layers, since potential attacks on the wireless layer could influence our requirements for a routing protocol. Furthermore, the effect that MaritimeManet may have on the likelihood or impact of certain attack paths will be taken into account in the risk analysis.

Other than the attacks themselves, the risk analysis also consists of the impacted pillar(s) of the CIA triad (confidentiality, integrity, or availability), the likelihood of successful exploitation and the impact that such exploitation would likely have. Likelihoods and impacts are scored on a 3-level scale ranging from low to high, and are relative to the most and least likely or impactful attack out of the ones identified. Risk levels are then derived by considering the likelihood and impact scores as numerical from 0 to 2 and adding them together, which results in a new score from 0 (--) to 4 (++).

An effort was made to make an informed risk analysis based on the importance of the pillars of the CIA triad, but depending on the exact use case of the network, a pillar may be more or less important than what was assumed. In reality, the actual scores might therefore deviate to some extent.

Note that this section does not serve as an exhaustive compilation of every attack path in existence; rather, it aims to identify attacks based on the unique network architecture, as well as the most common attacks found in literature, which other attacks are often based on. Other attacks can be found in literature, such as Sybil or disassociation attacks[23], but these either only apply to a few specific routing protocols, can be grouped under a common risk ('DoS') or can only be performed by exploiting one of the already mentioned attacks, such as impersonation. It should however give enough information to deduce the likelihood and impact of the attack paths that have not been listed here.

4.1 802.11

Since MaritimeManet uses 802.11s with SAE for its wireless security, we will perform a risk analysis of this protocol specifically. Table 1 illustrates a summary of the attack paths that were identified.

Denial-of-Service (DoS) attacks aim to compromise the availability of the network. In wireless networks, an attacker could intentionally cause packet collisions or generate radio interference at the operational frequency, an attack that is also known as jamming [23, 25]. While some mitigations — such as frequency hopping — are possible, DoS attacks are nearly impossible to completely prevent because there is a limited amount of unlicensed spectrum that may be used. Additionally, the control radio of MaritimeManet currently does not support frequency hopping. However, these attacks are highly physically targeted, and MaritimeManet nodes are spread out over a large physical area using directional antennas, which makes it infeasible to carry out such an attack on the entire network at once. This means that successful execution will likely first require

Table 1. 802.11s Risk Analysis (Likelihood / Impact / Risk)
(Scores are relative to each other)

ID	Attack	CIA	$\mathbf{L}_O$	$\mathbf{I}_O$	$\mathbf{R}_O$	$\mathbf{L}_M$	$\mathbf{I}_M$	$\mathbf{R}_M$
A0.0	DoS	A	+	+	++	+	-	o
A0.1	DoS - SAE	A	-	+	o	-	-	--
A0.2	Passive eavesdropping	C	+	o	+	-	-	--
A0.3	Wormhole	C / A	o	o	o	o	o	o
A0.4	MITM	C / I	-	+	o	-	+	o
A0.5	PSK brute-force	C / I	-	+	o	-	+	o
A0.6	SSID Confusion	I	?	?	?	?	?	?

identification of vulnerable nodes or routing paths that should be targeted. The impact therefore highly depends on how much information the attacker is able to gather about the physical topology of the network. Furthermore, it also depends on how quickly the used routing protocol is able to adapt to the sudden loss of connection to one of its neighbours, and on how many different next hops a node uses to route or forward traffic.

DoS attacks can also occur by leveraging implementation details of a protocol, rather than just the physical layer. In SAE, clients are required to do a substantial amount of work upon receipt of a Commit message during peering. This operation occurs before the other peer has been authenticated, so an attacker could flood a node with bogus peering attempts in order to overwhelm the processor of the node, causing a denial of service. This is also referred to as clogging [6, §12.4.6]. While SAE features a mitigation against this specific threat, it is worth noting that in WPA3-SAE, which also uses SAE, the mitigation has been bypassed before [32]. Other unauthenticated DoS attacks have also been found in different areas of the protocol, so it is not unlikely that undiscovered variations of these may still exist today [7].

One vulnerability that most wireless networks have in common is **passive eavesdropping**. While the content of each 802.11 packet is encrypted, some information that is necessary for the network to operate, such as the source and target MAC addresses and the MBSS identifier, are transferred in plain text [6, §14.2.2]. In a network with omnidirectional antennas, it is easy to snoop on the packets that are transferred over the air and deduce which nodes are exchanging packets with which patterns. But a MaritimeManet node has multiple antennas, each with its own unique MAC address that is more difficult to trace back to a single node. Do note that an attacker would still be able to obtain this information if the MBSS identifier of a mesh connection contains the source and/or target node. In any case, execution of this attack is significantly more difficult in MaritimeManet since an adversary would generally need to be physically situated directly in between two nodes in order to intercept their communication. This makes it very difficult to execute the attack on a large scale in the network.

There does exist a situation where an attacker does not need to be physically located between two nodes in order to record encrypted frames. If two nodes are out of reach of each other, an attacker could capture raw frames from each node and replay them to the other, making the nodes think that they are physically close to each other. This is called a **wormhole attack** and would lead to them establishing a connection via the attacker [23, 31]. This is essentially the 'active' counterpart to an eavesdropping attack. The attacker would still not be able to decrypt the nodes' traffic, but it would allow them to more effectively perform an eavesdropping attack. They could also attempt to selectively drop packets or (periodically) drop the connection entirely in order to attempt to destabilize the network. Similarly to a DoS attack, the impact of this attack heavily depends on how many packets nodes happen to route or forward over the 'fake' link.

SAE prevents a wormhole attack from turning into a **Man-In-The-Middle (MITM) attack**. When performing an MITM attack, the attacker will relay messages between two nodes while decrypting any packets that it receives [24]. This usually works by modifying the cryptographic keys that the nodes exchange, such that it is able to transparently decrypt incoming packets, read their contents and re-encrypt them before forwarding them to the destination node. This is not possible in MaritimeManet because SAE requires both nodes to prove its knowledge of the secret PSK, which the attacker presumably does not have. Additionally, each node uses the PSK and MAC addresses of both nodes in order to derive a shared seed, which will be different if the attacker poses as a different STA with a different MAC address [17]. For this reason, the adversary must forward packets unchanged if it wants nodes to establish a connection through its own link.

It is also not possible for the attacker to capture frames and **brute-force** the password offline. This is because SAE is based on zero-knowledge proofs, and the PSK itself is never shared over the link, not even in a hashed form. This is in stark contrast to WPA(2)-PSK, where such an attack is possible [7]. Compromising the confidentiality or integrity of SAE hinges on the computation of a discrete logarithm, which is currently regarded as being computationally infeasible [17]. While it is of course still possible to attempt a brute-force by repeatedly trying to peer with a node's sector(s), this is subject to heavy rate-limiting by the same process that mitigates the clogging DoS attack.

One final attack vector that is specific to SAE and is worth mentioning is an **SSID confusion attack**. SAE has two different methods to generate a shared seed between two peers: 'hunting-and-pecking' and 'hash-to-element' [17]. Hunting-and-pecking does not utilize the network's SSID to generate the seed, which means that it is possible for an attacker to spoof a network's MBSS identifier and make a node think that it is connected to a different network than it really is. Hash-to-element mode does not have this vulnerability, but is not (yet) enabled by default because it is a newer method, and not all devices support it yet [15]. This does still require another network that uses the same PSK, and requires the attacker to perform a wormhole-like interception in order to modify packets between nodes. While packet decryption is still not possible, as SAE evolves, it might be possible to downgrade the security of a connection if another node exists that does not yet implement those security upgrades. The likelihood and impact of this attack are difficult to

Table 2. Routing Risk Analysis (Likelihood / Impact / Risk)

ID	Attack	CIA	L	I	R
A1.0	Misrouting	A	+	o	+
A1.1	Packet dropping	A	+	o	+
A1.2	Black / Gray hole	A	o	+	+
A1.3	Impersonation (control)	I	o	+	+
A1.4	Impersonation (data)	I	o	+	+
A1.5	Replay attack	I / A	o	o	o
A1.6	Eavesdropping (control)	C	+	-	o
A1.7	Eavesdropping (data)	C	+	?	?

determine at this stage, because they highly depend on the DND component and sector configuration of MaritimeManet.

4.2 Routing

Compared to wireless standards, there are many more routing protocols to choose from in an ad-hoc network. This makes it likely that certain attack paths will only be viable in the implementation of a few specific routing protocols. Despite this, several attack paths can be identified that aim to attack the common principles of routing protocols, such as the sharing of routes with neighbours. Note that these attacks are inherently independent of the carrier technology, so no distinction is made between MaritimeManet and a network with omnidirectional antennas. Furthermore, the likelihood and impact scores as indicated in Table 2 are specific to the attacks themselves, and for determination of these scores it is assumed that no security is provided by any of the lower layers. This way, no assumptions are made about the possibility of chaining attacks, and it becomes possible to analyse both layers separately from each other. But in real life, these routing attacks may only be executed if and only if:

- The security of the wireless layer is breached, or;
- A legitimate node is compromised.

One attack path that is applicable to almost any routing protocol is **misrouting**. In misrouting, a malicious node will intentionally forward packets to the wrong neighbour. This may simply increase the overall latency of the network, but also cause routing loops [29]. Packets that utilize IPv4 or IPv6 do contain a TTL or Hop Limit value, which will eventually break routing loops by limiting the number of times the packet can be forwarded [3, 19]. Despite this, packets that travel in loops may still take up a considerable amount of the available bandwidth and impact the total throughput of the network.

An attack similar to misrouting is **packet dropping**. A packet dropping attack entails that the adversary will intentionally drop packets that it should be forwarding instead. This is usually done selectively; for example, the attacker may choose to forward all control plane packets correctly, but drop all data packets. This may make other nodes think that the path is 'up', but still prevent communication [21, 29]. This attack can be difficult to detect because there exist legitimate reasons for packets to be dropped, such as poor link quality, low routing capacity or buffer overflows. Still, the impact is typically limited to paths that actively travel through the adversary.

Packet dropping can be made more effective by combining it with a **black- or gray hole attack**. In a such an attack, a malicious node will emit false routing information to its neighbours in order to 'pull' traffic towards them. By telling other nodes that it has a low-cost path to node X, it may convince them to forward more traffic towards itself. Once done, the adversary may start dropping some or all data packets in order to degrade the performance of the network [23, 25]. This attack can seriously disturb the routing capabilities of the network, but is comparatively more difficult to perform.

Impersonation is another well-known attack path in routing protocols. The goal of an impersonation attack is to send messages that appear to originate from a different node than the attacker. On the control plane this is often combined with another attack, such as a black hole attack, in order to strengthen its effects. Additionally, nodes may be able to receive routing packets that are addressed to the node they are posing as [29].

Impersonation may be applied to the data plane as well: for example, an adversary may attempt to assume other nodes' identities by spoofing IP addresses in order to attack upper layers [13, 29]. This attack is generally more difficult to perform than e.g. simple packet dropping, so the likelihood is scored slightly lower. However, if applications that use the network do not implement any authentication themselves, the impact of this attack can be much higher, and it could be used as a gateway towards more sophisticated attacks on these higher layers.

One attack path that may not be as well-known as the others is a **replay attack**. The idea of this attack is to record legitimate routing updates sent by other nodes in the network, and periodically re-emit them to your neighbours. This may make the network less responsive to topological changes [16]. The advantage of this attack is that this strategy is much more subtle and may be harder to detect than simple packet forging. To combat this, routing protocols will have to take the 'freshness' of updates into account, which ensures that nodes will reject routing packets which they have previously received, or those that are less recent than the message they last received [26, 31]. While being relatively easy to exploit in a typical routing protocol, its impact highly depends on the topology of the network and on how quickly it changes.

Eavesdropping is not only an attack path in 802.11, but in routing protocols as well. This attack can be applied to both the control and data plane, and the impact of the attack differs per strategy. When applied to the control plane, an attacker intends to gather data about the topology of the network [13]. This is similar to the passive eavesdropping attack on 802.11, but the nature of routing protocols means that they generally exchange much more topological information. This attack could therefore be used to prepare a more targeted attack on individual nodes. Still, it is worth noting that even though it is very easy for an attacker to capture unencrypted packets flowing through its interfaces, many routing protocols do not 'expose' the entire network topology to a node, but only the best or fastest path(s) to other nodes.

When applied to the data plane, the impact of an **eavesdropping** attack highly depends on the application(s) that will be utilizing the network. Some applications may be exchanging data in plain text, which will then be visible to the attacker if nodes are routing packets via them. It is also likely that other layers, such as the transport layer, will be exposed. While routing protocols may encapsulate and encrypt data packets in order to mitigate this, it is also possible to utilize an encrypted peer-to-peer VPN solution such as WireGuard [12] or MACsec [4].

4.3 Discussion

From Table 1 and 2 we can see that there are many attack paths that an adversary may attempt to take. When inspecting the 802.11s risk analysis for MaritimeManet, we can see that the attacks with the highest impact have the lowest likelihood. This makes sense and is a good sign: 802.11(s) has had security features engineered into the protocol from the start, which mitigate the most important attacks. The risk analysis also shows that there currently does not appear to be a way to make a significant impact on the confidentiality or integrity of the wireless layer. While some information such as MAC addresses and MBSS identifiers may still be obtained, this does not appear to have the potential to affect the network in any meaningful way. Additionally, MaritimeManet's directional antennas do not appear to have a negative effect on the likelihood or impact of the analysed attack paths, and it even decreases the potential risk for some attacks.

When we look at the risk analysis for routing protocols in Table 2, a stark contrast can be observed compared to our 802.11 analysis: risk scores are significantly higher. The main reason for this is that this analysis was carried out while assuming that the attacker has full network access, either through a wireless vulnerability or through hijacking of a node. While this may sound unrealistic, it was in fact necessary in order to create a more accurate risk analysis. Otherwise, the analysis would not only require knowledge about the likelihood of wireless attack vectors, but also about the likelihood of a node being hijacked, or the likelihood and impact of software, hardware or human errors that grant network access in other ways. Leaving this open in the analysis allows people to make their own conclusions about this matter depending on the use case of the network.

5 SECURITY REQUIREMENTS

Given the risk analyses for both 802.11s and common routing processes, we are able to draft some requirements that a secure routing protocol should have in order to mitigate the outstanding risks. The aim of these requirements is to protect against the case where either the wireless security is breached, or a legitimate node is hijacked. In particular, it does not aim to protect against the scenario where nodes are directly communicating with an (unknowingly) hijacked node. Rather, the goal is to prevent such nodes or other adversaries from impacting the confidentiality, integrity, or availability of the traffic between any two other, non-breached nodes, as long as such a non-breached path between those nodes exists.

Note that not all attack paths that were identified can be fully mitigated on a routing level: for example, passive eavesdropping on the 802.11 layer and SSID confusion attacks are unaffected by the choice of routing protocol. Both of these attack paths should instead be mitigated in the configuration of the wireless network. PSK brute-force and Man-in-the-middle attacks are left out for similar reasons.

It is worth mentioning that the drafted requirements do not fully adhere to the typical S.M.A.R.T. criteria. This is because not all security requirements can be easily quantified, and sometimes the goal is to do something as good as possible rather than to hit a specific goal. Additionally, the scope of each requirement has been mostly limited to one specific technical pillar; for example, in cryptography, encryption and signing of data accomplish two distinctly separate goals, so the requirements asking for those goals are separate as well.

The requirements mentioned in section 5 have been summarized in Table 3. The 'importance' column in this table roughly indicates how important implementation of said requirement is, and is relative to the least or most important requirement out of the ones identified. The score is based on the risk scores of the attack(s) that the corresponding requirement mitigates. In the case of 802.11, the specific score for MaritimeManet was used.

One of the most risks with the wireless layer is the presence of DoS attacks. There are many ways for an attacker to inhibit the proper functioning of a wireless link between two nodes, and the nature of MaritimeManet's operational environment means that we should prepare for the sudden loss of a previously functional link. One of the ways this could be prepared for is **multipath routing (R0)**: if possible, a node should keep track of multiple paths to reach a given destination, so that it is not fully reliant on a few other nodes in order to reach a target node [22]. This reduces the impact of a sudden link loss. It also increases the connectivity of the network: the minimum number of links that need to be removed to isolate a part of the network. This requirement may also reduce the impact of wormhole attacks, as it would become less likely for a node to route all of its traffic over the malicious link.

Besides forwarding diversity, **convergence of routing paths (R1)** after a topology change should finish quickly; in other words, the algorithm's convergence time should be low. This minimizes the time it takes for a node to find new routing paths to other nodes, and therefore increases the resilience against the loss of a link due to natural causes or a DoS attack. Additionally, this may reduce the impact of a sudden disconnect during a wormhole attack.

Third, a node should **incorporate the packet loss ratio of a link as a metric for deciding its routing paths (R2)**. This reduces the impact of misrouting, packet dropping and black / gray hole attacks by malicious or compromised nodes. By effectively reducing the probability of unreliable paths from carrying packets, malicious nodes may be less enticed to deliberately disturb traffic flows. After all, doing so might cause them to lose an advantageous topological position in the network.

In order to prevent malicious nodes from modifying other nodes' routing packets, **the authenticity and integrity of these messages must be secured (R3)**. This principle is also called non-repudiation, and completely inhibits impersonation on the control plane. Implementation of this requirement will allow nodes to verify that a routing packet has not been tampered with and was truly sent by the node that the message claims to originate from. Usually, this process involves public key cryptography, and will require each

Table 3. Security requirements for routing protocols

ID	Mitigates attack(s)	Requirement	Importance
R0	A0.0, A0.1, A0.3	Packets should be able to reach their destination using multiple paths.	Med
R1	A0.0, A0.1, A0.3	Routing paths should converge quickly after a topology change.	Med
R2	A1.0, A1.1, A1.2	Paths with high packet loss ratios should have a lower priority for routing and forwarding.	High
R3	A1.3	It must be possible to verify the authenticity and integrity of routing packets.	High
R4	A1.4	It must be possible to verify the authenticity and integrity of data packets.	High
R5	A1.5	Routing packets must be verifiably live.	Med
R6	A1.6	The confidentiality of routing packets must be secured.	Med
R7	A1.7	The confidentiality of data packets must be secured.	?

participating node to explicitly ’trust’ other nodes before being able to participate in the network.

Non-repudiation should also be applied to the data plane (R4). In order to prevent impersonation attacks such as IP spoofing, the receiving node will need to be able to verify the authenticity and integrity of packets it receives. The process to facilitate this would likely be similar to how non-repudiation for the control plane is handled. It is worth noting that unlike the control plane, the data plane carries higher layers of traffic, so it is also possible to implement this requirement in a higher layer. For example, a peer-to-peer VPN solution could be used, or it could be directly implemented in the application layer.

Other than non-repudiation, **routing packets must also be verifiably live (R5)**. This means that nodes must be able to verify that such messages are still recent and are not being repeated. This mitigates routing replay attacks, since the attacker will be unable to re-emit old routing packets without being detected.

A welcome security requirement for routing protocols would be **encryption on the control plane (R6)**. By protecting the confidentiality of routing packets, attackers that manage to breach the security of the wireless layer will have a harder time gathering topological network information. This will make it more difficult for them to launch further, more targeted attacks. Note that this requirement cannot protect against insider attacks by breached nodes: since routing algorithms inherently need to share a lot of information with neighbours, insiders would still be able to capture routing information, since the previously-legitimate node would have needed to decrypt it.

Finally, **encryption on the data plane (R7)** can be introduced to mitigate eavesdropping. Similarly to the non-repudiation requirement, this can also be solved using a peer-to-peer VPN on higher layers. Depending on the application(s) that will be using the network, it may also not be necessary to encrypt all data plane traffic; for example, if applications are already using an encrypted communication layer to communicate. This makes it difficult to judge the potential impact of this attack.

6 ROUTING PROTOCOLS

Based on the selected security requirements, we are able to compare routing protocols to evaluate how well they protect against the potential attack paths. This comparison has been summarized in Table 4. R1, convergence time, has been scored on a low — high scale instead of yes or no; this score is not an absolute claim and is relative to the lowest or highest performing routing protocol out of the ones listed. Additionally, this metric is only available for some protocols, and is sometimes based on the non-security oriented version of the protocol; it should therefore only be used as an indication. Other than the security requirements, in order to estimate the usability of these routing protocols, some functional requirements have also been drafted to compare against.

The protocols to compare have been chosen based on relative popularity in literature or due to their unique properties. The reason for this is that the pre-existence of an OpenWrt package for a protocol is a hard requirement.

6.1 Functional properties

Besides the various security requirements, there are a few functional properties of routing protocols that we would also like to analyse in our comparison.

- **Layer**: routing protocols generally run on layer 2 (data link) or layer 3 (network). Since devices already ship with MAC addresses from the factory, protocols running on layer 2 generally do not require further network configuration in order to function. When running on layer 3, it may be necessary to set up (static) IPv4 or IPv6 addressing first.
- **Link-state / Distance-vector routing**: routing protocols can be broadly categorized into these two categories. In distance-vector (DS) routing, nodes only exchange information about paths between them and their immediate neighbours. In link-state (LS) routing, nodes exchange the state of the entire network with each other. Some protocols are a combination of these two.
- **Proactive / Reactive**: Proactive routing protocols determine paths to other nodes in the network in advance, while reactive protocols will only do so once they need to send a packet to the target node.
- **OpenWrt implementation**: Not all routing protocols actually have a functional implementation; some have only been worked out in literature. The availability of a routing protocol in OpenWrt is a hard requirement.
- **Multicast support**: IP multicast is a feature that allows a sender to address a single IP packet to multiple listeners. Not all protocols support multicast, and unfortunately it is not always clear whether a given protocol supports it. Multicast support is a desirable feature for choosing a routing protocol.

Table 4. Properties and requirement adherence of several routing protocols

Protocol	Layer	LS / DV	Pro- / Reactive	OpenWrt	Multicast	R0	R1	R2	R3	R4	R5	R6	R7
ARAN [30]	?	DV	Reactive	No	?	-	?	-	+	-	+	-	-
Babel HMAC [11]	3	DV	Proactive	No	?	?	Med	?	+	-	+	-	-
batman-adv [11]	2	Both	Proactive	Yes	Yes	-	?	-	-	-	-	-	-
BMX7 [26]	3	Both	Proactive	Yes	No	?	High	?	+	-	+	-	-*
OLSR [9]	3	LS	Proactive	Yes	No	-	Low	-	-	-	-	-	-
PASER [31]	?	DV	Reactive	No	?	?	?	?	+	-	+	-	-

6.2 Comparison

ARAN [30] was one of the first routing protocols that attempted to implement some form of security. It was created in early 2005 and facilitates non-repudiation and replay attack protection on the control plane, but makes no promises about security on the data plane due to performance concerns. Paths are chosen by 'requesting' a node and seeing which path responds first — this provides very basic protection against packet drops by adversaries, but only if said adversary also drops routing packets. The biggest downside of ARAN is that nodes cannot operate completely independently: it requires the presence of a trusted certificate server, which is assumed to be regularly contactable. A solution to this problem has been proposed by utilizing DHCP to request public keys, but this comes with new security issues and no implementation of this mechanism has been created so far. An open-source implementation of ARAN for Linux 2.4+ is supposed to exist, but at the time of writing, the download for this appears to be unavailable.

Babel HMAC [11] is a modified version of the popular distance-vector Babel [8] routing protocol. It aims to add authentication to the existing Babel messages, ensuring non-repudiation and protection against replay attacks. Routing packets are assumed to be non-confidential, and no data plane protection is provided. The standard explicitly does not define a predefined metric function for routing paths, so this aspect is left up to the implementation. Currently, no practical implementation of this protocol appears to exist.

Batman-adv [20] is the protocol that is currently used in test setups for MaritimeManet. It belongs to the B.A.T.M.A.N. family of routing protocols which are optimized for MANETs, but batman-adv currently does not implement any security features. Several versions of the algorithm exist, but the current default version, BATMAN IV ('version 4'), uses originator messages (OGMs) and other signals to approximate link quality. This is then used to compute a path metric. This is useful for wireless mediums, but similarly to ARAN, this does not protect against selective packet dropping.

BMX7 [26] also belongs to the B.A.T.M.A.N. family. It is very closely related to the high-performance BMX6 routing protocol, but adds a layer of individually-trusted security. Routing packets are unencrypted but cryptographically signed, which prevents modifications by anyone other than the author of the packet. The idea of the protocol is that nodes will announce which other nodes are allowed to carry traffic that is addressed to them in order to create a personal trusted circle. BMX7 is fully ad-hoc, IPv6-only and runs on layer 3, so it also manages IP addresses. These addresses are derived from the hash of a node's public key, which ensures non-repudiation. While no data-plane security promises are made, it does ensure that impersonation cannot lead to the poisoning of routing paths, which severely limits its scope. Furthermore, because data packets only travel through nodes that are trusted by their recipient, their integrity can only be compromised by hijacking a legitimate node. There does currently exist an experimental BMX7 plugin that adds WireGuard-based end-to-end encryption between nodes in the mesh, but it is not (yet) ready for production use [28].

OLSR [9] and its younger sibling OLSRv2 [10] are popular link-state routing protocols. Both of these protocols do not feature any security considerations whatsoever. Paths are selected by running a shortest path algorithm on a directed graph of the network, which nodes obtain by exchanging extensive topological information with each other. Compared to Babel and BMX6, OLSR suffers from a high convergence time when the network topology changes often [27].

The final protocol we will discuss is **PASER** [31]. The interesting aspect of this protocol is that it defends against wormhole attacks by incorporating the physical geolocation of nodes in its routing packets. Other than that, in terms of security features, it is very similar to ARAN and makes the same assumption about a pre-existing centralized public key infrastructure (PKI). While unclear from the paper, PASER does not appear to implement any data-plane security measures. Development of this protocol also does not appear to have progressed past the simulation stage.

6.3 Discussion

The security requirements and analysis of routing protocols tell us that the security-oriented protocols provide very similar levels of security. All of them aim to protect the control plane from attackers, and consider data plane security to be out of scope. At first sight, it appears that these protocols only implement one out of the three high-importance requirements. However, assuming that data-plane security can be achieved using peer-to-peer VPN connections between nodes in the mesh, they still provide a significant amount of security. That does still leave Requirement 2, but there was insufficient information available to make informed statements about this aspect.

7 IMPLEMENTATION

In order to test the feasibility of using a secure routing protocol in MaritimeManet, a test setup was created. Initially, testing was performed on physical Alix 2d2 [1] devices as a proof of concept. However, this setup scaled rather poorly with the available hardware, so to facilitate more efficient testing a virtual environment was set up using Proxmox VE [2].

7.1 Implementation environment

After installing Proxmox VE on a dedicated machine, it becomes possible to create and run many virtual machines on that host. Each virtual machine acts as a node and runs the entire OpenWrt operating system with its own Linux kernel, which allows for a realistic technical simulation. Given an OpenWrt x86-64 image, a template VM can be created by attaching it as the only disk and adding a virtual serial port for terminal access. This template VM can then be easily cloned to a new VM in order to create a new, fresh node. For these experiments, OpenWrt 23.05.3 was used.

For networking, a VLAN-aware Linux bridge was created in Proxmox to attach the VMs to. In the configuration of a VM, it is possible to attach a network interface that is connected to said bridge. The interface will then show up in OpenWrt as a simple Ethernet device. Because not all VMs should be directly connected to each other, a VLAN tag should be set on the interface from within Proxmox, which restricts interfaces with the same tag to the same virtual network segment. For example, in order to connect VM '02' and '04' together, both should have a network interface with the same VLAN tag, such as '0204'. This VLAN tagging is completely transparent and invisible to the guest OpenWrt OS. A schematic diagram of the described environment can be seen in Figure 3.

7.2 Technical details

To facilitate easy deployment of multiple VMs, a custom build script was created that uses OpenWrt's image builder to compile the routing protocol into the final image. This image can then be easily deployed to multiple VMs. Tools such as `tcpdump` and `iperf3` were also installed for debugging purposes.

When attaching a network interface from Proxmox, OpenWrt will create internal interfaces named `eth0`, `eth1`, and so on. Therefore, it is important to pre-configure the routing protocol to use these interfaces, even if they do not exist yet. This allows hotplugging of the interfaces from Proxmox in order to test dynamic topology changes.

8 EVALUATION

After comparing various different routing protocols in Section 6, it was decided to test BMX7 in order to see how suitable it is for use in MaritimeManet. BMX7 was chosen because it is the only security-oriented protocol out of the ones analysed that has an implementation for OpenWrt. The performance of BMX6 is also supposedly very good compared to Babel and OLSR under various circumstances [27, 33], so the hope is that this performance is present in BMX7 as well.

BMX7 can be installed by compiling the `bmx7` OpenWrt package into an image as described in Section 7.2. It must be configured to use the `ethX` interfaces, which can be done by modifying the configuration file from OpenWrt in `/etc/config/bmx7`. The following lines should be added for each interface:

```
config 'dev' 'mesh_1'
    option 'dev' 'ethX'
```

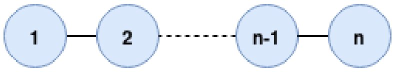

Fig. 2. Evaluation topology.

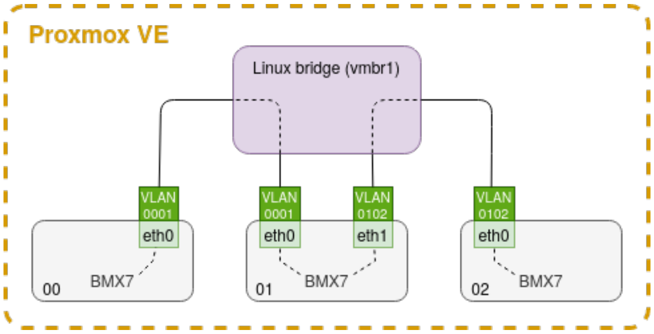

Fig. 3. Schematic overview of Proxmox setup.

It may be necessary to 'up' a new interface using the command `ip link set ethX up`. BMX7 will automatically generate a /128 IPv6 address for the node it is running on and apply it to the indicated interfaces. However, no communication will be possible between nodes except for immediate neighbours. This is because, for a node to receive traffic from another node, they must either be direct neighbours, or there must exist a path of nodes between the sender and recipient node which are all trusted by the recipient. This trust must be manually configured for each node, and this has not yet been done. For testing purposes, the following bash command may be used to automatically trust all nodes in the network, which will enable communication:

```
for i in $(bmx7 -c topology | awk '{ print $1
    }' | tail -n +3 | uniq); do bmx7 -c --
    setTrustedNode $i; done
```

8.1 Multicast support

During analysis it was unclear whether BMX7 supported multicast or not, so this was one of the things that was desirable to test. In IPv4 and IPv6, sending a packet to a multicast address should cause it to be received by all other devices in the network. We can test for multicast support by sending an ICMP echo request (ping) packet to a multicast address. Nodes that receive a ping packet will, by default, respond with an ICMP echo reply. In a multi-hop network topology, we therefore expect to receive exactly one response from each node; if we only receive responses from our immediate neighbours or no responses at all, we can say that multicast is not supported.

For this test, the evaluation topology (Figure 2) will be used with $n = 10$ nodes. First, we will send a regular unicast ping from one of the ends to the other end using the following command: `ping -6 -c 1 <IPv6 address>`. This will verify that all links are functional, and is a prerequisite for starting the test. We expect to receive exactly one response. After this, the following command is used to test a multicast ping: `ping -6 -c 1 ff02::1`. The test is successful if and only if the second command returns `n` responses, where `n` is the number of nodes in the network, and the node on the other end of the network has responded exactly once. Note that we will also receive a response from our local machine.

After conducting the above experiment, we can say that BMX7 does not support multicast routing. While both ends of the network are able to reach each other, sending ICMP echo packets to a multicast address only results in responses from immediate neighbours.

8.2 IPv4/6 Tunnelling

In BMX7, each node has one full /128 IPv6 address that can be used to reach it. But in a real-life scenario, this is not enough; if each node represents a single ship, there will be devices on the ship that require their own IP address. These devices need to be able to communicate with devices in networks of other ships. BMX7 solves this by using 4in6 or 6in6 tunneling, which encapsulates IPv4 or IPv6 packets in another IPv6 packet. This outer IPv6 packet is routed by the BMX7 mesh to the correct 'gateway' node, which then strips this outer header from the packet in order to route it further in the network. Practically speaking, this works by telling a BMX7 node to 'advertise' a subnet which it wants to route, and then telling other nodes to 'accept' this advertisement. Note that although this concept is called tunnelling, the tunnel is not actually encrypted. There is also no active tunnel 'connection,' as it is completely stateless: there is no active communication happening between tunnel endpoints, other than the encapsulated packets. All information to set up the tunnel is exchanged via the routing protocol [5].

We will test this by using the evaluation topology (Figure 2) with $n = 10$ nodes. Both ends of the network will advertise a subnet, `10.10.0.1/24` and `10.10.9.1`, to the network. The last octet, 1, is the 'default gateway' of the subnet and is assigned to the mesh nodes themselves. This can be configured using the following commands, where `X` is the third octet (0 or 9) of the subnet that should be advertised by the node:

```
bmx7 -c tunDev=Default /tun4Address=10.10.X
    .1/24
bmx7 -c tunOut=v4Default /network=0.0.0.0/0
```

The first command begins advertising a /24 IPv4 subnet to the mesh, while the second command instructs the node to 'accept' any subnet that is advertised by any other node. A subnet is only reachable by a node after it has accepted it. After this, we will ping `10.10.X.1` from both nodes to test whether IPv4 tunnelling works. If it does, the experiment is successful.

After attempting the above experiment, both nodes were successfully able to ping each other's addresses, so IPv4 tunnelling works. The experiment was successful.

9 FURTHER WORK

There are not many secure OpenWrt-compatible protocols available, and while BMX7 appears to provide the level of security that is desired, it does have some practical shortcomings: notably, having to manually trust each node in the network and the lack of multicast support. Ideally, these issues should be overcome before the protocol is used in MaritimeManet, but more research is required to determine a good solution to this new problem. Example solutions could include running a process on each node which adjusts BMX7's settings on-the-fly, or further development of the protocol to better suit MaritimeManet's needs.

10 CONCLUSIONS

It is clear that there are various attack paths that an adversary may attempt in order to compromise the confidentiality, integrity, or availability of MaritimeManet. However, compared to a network with omnidirectional antennas, it appears that directional antennas provide a significant improvement when it comes to the risks that these attack paths pose. Particularly, denial of service and passive eavesdropping attacks are less likely to occur and have a lower impact on the network as a whole.

From a technical aspect, given that 802.11s security is in place, there currently does not appear to be an effective way to impact the confidentiality or integrity of the network as an outsider. However, there is more to risk assessment than technical attack paths. Legitimate nodes may be hijacked, and security vulnerabilities may be present in the software that is used, leading to a vulnerable network. In order to protect the network in these situations, the most important attack paths from the routing protocol risk analysis, such as misrouting and impersonation attacks, must be mitigated. This can be achieved by using the security requirements from Table 3 as a guideline.

In order to test a routing protocol for use in MaritimeManet, a Proxmox environment with virtualized OpenWrt instances turned out to be an efficient way to quickly perform experiments with such a protocol. When using custom-built images, this setup allows rapid deployment of various network topologies. This significantly reduces testing complexity while still being realistic in a technical aspect.

From a security perspective, it was found that BMX7 could be a good option for implementation in MaritimeManet. It provides good coverage of the drafted security requirements, and most of the ones it does not implement may be achieved through other means. It also has a very well functioning OpenWrt implementation. However, it does not support IP multicast, and relies heavily on the configuration of individual nodes. This last issue may require additional measures for network management in a practical deployment of MaritimeManet.

From a practical perspective, batman-adv appears to be the best option out of the ones analysed. It is a battle-tested protocol that is optimized for wireless networks, has an implementation in OpenWrt, and has excellent support for multicast.

Which routing protocol should be used will ultimately depend on the acceptable level of risk in MaritimeManet. While the network should already be 'safe enough' with 802.11s from a theoretical standpoint, there do still exist practical ways for the wireless security to be impacted, in which case a secure routing protocol could provide extra protection. On the other hand, using a protocol such as batman-adv instead of BMX7 brings many practical advantages. It might be possible to close this gap by further extending BMX7 such that it becomes more practical for MaritimeManet's purposes. But until then, choosing a protocol will mean making a trade-off between practicality and security.

REFERENCES

[1] [n. d.]. PC Engines alix2d2 product file. https://www.pcengines.ch/alix2d2.htm

[2] [n. d.]. Proxmox Virtual Environment. https://www.proxmox.com/en/proxmox-virtual-environment/overview

[3] 1981. *Internet Protocol.* Request for Comments RFC 791. Internet Engineering Task Force. https://doi.org/10.17487/RFC0791 Num Pages: 51.

[4] 2018. IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security. *IEEE Std 802.1AE-2018 (Revision of IEEE Std 802.1AE-2006)* (Dec. 2018), 1–239. https://doi.org/10.1109/IEEESTD.2018.8585421 Conference Name: IEEE Std 802.1AE-2018 (Revision of IEEE Std 802.1AE-2006).

[5] 2019. GitHub: bmx7/doc/Tunneling.md. https://github.com/bmx-routing/bmx7/blob/9020896f89006bc5d3487222eefc7ddea9e8b2bd/doc/Tunneling.md

[6] 2021. IEEE Std 802.11™-2020, IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2021). https://doi.org/10.1109/IEEESTD.2021.9363693

[7] Efstratios Chatzoglou, Georgios Kambourakis, and Constantinos Kolias. 2022. How is your Wi-Fi connection today? DoS attacks on WPA3-SAE. *Journal of Information Security and Applications* 64 (Feb. 2022), 103058. https://doi.org/10.1016/j.jisa.2021.103058

[8] Juliusz Chroboczek. 2011. *The Babel Routing Protocol.* Request for Comments RFC 6126. Internet Engineering Task Force. https://doi.org/10.17487/RFC6126 Num Pages: 45.

[9] Thomas H. Clausen and Philippe Jacquet. 2003. *Optimized Link State Routing Protocol (OLSR).* Request for Comments RFC 3626. Internet Engineering Task Force. https://doi.org/10.17487/RFC3626 Num Pages: 75.

[10] Christopher Dearlove and Thomas H. Clausen. 2014. *Optimized Link State Routing Protocol Version 2 (OLSRv2) and MANET Neighborhood Discovery Protocol (NHDP) Extension TLVs.* Request for Comments RFC 7188. Internet Engineering Task Force. https://doi.org/10.17487/RFC7188 Num Pages: 16.

[11] Clara Do, Weronika Kolodziejak, and Juliusz Chroboczek. 2021. *MAC Authentication for the Babel Routing Protocol.* Request for Comments RFC 8967. Internet Engineering Task Force. https://doi.org/10.17487/RFC8967 Num Pages: 17.

[12] Jason A. Donenfeld. 2017. WireGuard: Next Generation Kernel Network Tunnel. In *Proceedings 2017 Network and Distributed System Security Symposium.* Internet Society, San Diego, CA. https://doi.org/10.14722/ndss.2017.23160

[13] El-Sayed M El-Rabaie and Nancy A Al-Shaer. 2016. A Survey on Ad Hoc Networks. (Nov. 2016).

[14] Wojciech Galuba, Panos Papadimitratos, Marcin Poturalski, Karl Aberer, Zoran Despotovic, and Wolfgang Kellerer. 2010. Castor: scalable secure routing for ad hoc networks. In *Proceedings of the 29th conference on Information communications (INFOCOM'10).* IEEE Press, San Diego, California, USA, 2829–2837.

[15] Héloïse Gollier and Mathy Vanhoef. 2024. SSID Confusion: Making Wi-Fi Clients Connect to the Wrong Network. (2024).

[16] Priyanka Goyal, Vinti Parmar, and Rahul Rishi. 2011. MANET: Vulnerabilities, Challenges, Attacks, Application. 11 (2011).

[17] Dan Harkins. 2008. Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks. In *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008).* IEEE, Cap Esterel, France, 839–844. https://doi.org/10.1109/SENSORCOMM.2008.131

[18] Guido R. Hiertz, Dee Denteneer, Sebastian Max, Rakesh Taori, Javier Cardona, Lars Berlemann, and Bernhard Walke. 2010. IEEE 802.11s: The WLAN Mesh Standard. *IEEE Wireless Communications* 17, 1 (Feb. 2010), 104–111. https://doi.org/10.1109/MWC.2010.5416357

[19] Bob Hinden and Steve E. Deering. 1998. *Internet Protocol, Version 6 (IPv6) Specification.* Request for Comments RFC 2460. Internet Engineering Task Force. https://doi.org/10.17487/RFC2460 Num Pages: 39.

[20] Lang Jean-Philippe. [n. d.]. Doc-overview - batman-adv - Open Mesh. https://www.open-mesh.org/projects/batman-adv/wiki/Doc-overview

[21] Issa Khalil. 2008. MIMI: Mitigating Packet Misrouting in Locally-Monitored Multi-Hop Wireless Ad Hoc Networks. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference.* IEEE, New Orleans, LA, USA, 1–5. https://doi.org/10.1109/GLOCOM.2008.ECP.154

[22] S.-J. Lee and M. Gerla. 2001. Split multipath routing with maximally disjoint paths in ad hoc networks. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, Vol. 10. IEEE, Helsinki, Finland, 3201–3205. https://doi.org/10.1109/ICC.2001.937262

[23] Martin Andreoni Lopez, Michael Baddeley, Willian T. Lunardi, Anshul Pandey, and Jean-Pierre Giacalone. 2021. Towards Secure Wireless Mesh Networks for UAV Swarm Connectivity: Current Threats, Research, and Opportunities. http://arxiv.org/abs/2108.13154 arXiv:2108.13154 [cs].

[24] Ahmed M Al Naamany, Ali Al Shidhani, and Hadj Bourdoucen. 2006. IEEE 802.11 Wireless LAN Security Overview. (2006).

[25] T. Naeem and K. K. Loo. 2009. Common security issues and challenges in wireless sensor networks and IEEE 802.11 wireless mesh networks. (2009). http://bura.brunel.ac.uk/handle/2438/3662 Accepted: 2009-09-30T13:31:57Z Publisher: Advanced Institute of Convergence IT.

[26] Axel Neumann, Ester Lopez, Llorenc Cerda-Alabern, and Leandro Navarro. 2016. Securely-entrusted multi-topology routing for community networks. In *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS).* 1–8. https://ieeexplore.ieee.org/document/7429050

[27] Axel Neumann, Ester López, and Leandro Navarro. 2015. Evaluation of mesh routing protocols for wireless community networks. *Computer Networks* 93 (Dec. 2015), 308–323. https://doi.org/10.1016/j.comnet.2015.07.018

[28] Harry Pantazis. 2019. BMX7 - WireGuard Tunneling: Final Report. https://blog.freifunk.net/2019/08/26/bmx7-wireguard-tunneling-final-report/

[29] S A Razak, S M Furnell, and P J Brooke. 2004. Attacks against Mobile Ad Hoc Networks Routing Protocols. (2004). https://api.semanticscholar.org/CorpusID:15215602

[30] K. Sanzgiri, D. LaFlamme, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer. 2005. Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communications* 23, 3 (March 2005), 598–610. https://doi.org/10.1109/JSAC.2004.842547 Conference Name: IEEE Journal on Selected Areas in Communications.

[31] Mohamad Sbeiti, Niklas Goddemeier, Daniel Behnke, and Christian Wietfeld. 2016. PASER: Secure and Efficient Routing Approach for Airborne Mesh Networks. *IEEE Transactions on Wireless Communications* 15, 3 (March 2016), 1950–1964. https://doi.org/10.1109/TWC.2015.2497257

[32] Mathy Vanhoef and Eyal Ronen. 2020. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In *IEEE Symposium on Security & Privacy (SP).* IEEE.

[33] Roger Baig Viñas. 2014. Evaluation of Dynamic Routing Protocols on Realistic Wireless Topologies. (Sept. 2014).

A USE OF AI AND OTHER TOOLS

During the preparation of this work, the author(s) used LanguageTool in order to verify the spelling, style and grammar of the text contained within the work. LanguageTool was used to correct small typographical mistakes in the text of this work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

During the preparation of this work, the author(s) used Kagi, Google Scholar, IEEE Xplore and other search engines in order to gather relevant sources for this work. Search engines may use artificial intelligence to determine the relevance and order of search results, potentially impacting which references were used in this work. After using these tools/services, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

During the preparation of this work, the author(s) used Zotero in order to collect and store copies of references that were used in this work. Zotero may use artificial intelligence in order to collect bibliographical details from used references. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

During the preparation of this work, the author(s) used diagrams.net in order to create Figure 2 and Figure 3 in this work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

During the preparation of this work, the author(s) used Overleaf in order to write the LaTeXsources that correspond to this work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.