Integrating Trust Establishment into Routing Protocols of Today's MANETs

Alexander Oberle and André Rein
and Nicolai Kuntze and Carsten Rudolph
Fraunhofer SIT (Germany)
{oberle|rein|kuntze|rudolphc}@sit.fraunhofer.de

Janne Paatero and Andrew Lunn
and Peter Racz
RUAG Switzerland Ltd.
{janne.paatero|andrew.lunn|peter.racz}@ruag.com

***Abstract*—Conventional network protocols and its security mechanisms fail to cope with arising challenges in trust. Well known concepts from the domain of Trusted Computing can be applied to the example of mobile ad-hoc networks (MANETs) in order to establish extended trust capabilities between devices. The approach of such an anchor of trust in MANETs shows interesting possibilities since no central instances such as Access Points are involved in those networks. The communication between directly connected devices of the network is protected by a cryptographic protocol making use of a Trusted Platform Module (TPM) that serves as root-of-trust on each device. Such a hardware chip allows devices to attest the local system state and assess states of remote systems. Building on this, transmission of routing and payload data can be restricted to devices in trustworthy states. The resulting mobile ad-hoc network, by using this protocol, is protected against many of today's security threats. Single malicious devices are automatically recognised and excluded from participation in the network by all devices. Especially the dissemination of misleading routing information, which affects the availability of the whole network, is effectively prevented by the developed protocol. Thus, it is shown that the device itself is secured by a hardware TPM. Also the communication is secured, by verifying the device's state between the counterparts.**

I. Introduction

Mobile devices, as they are used by civil protection organisations, must be versatile despite being used in potentially malicious environments. Especially mobile ad-hoc networks (MANETs) are highly vulnerable to existing threats (such as routing attacks or compromising a system and starting insider attacks on the network), since they need to be flexible and operate without strongly protected infrastructural systems. Immediate exclusion of attackers or compromised systems injecting falsified data, is crucial in a network where all participants act as routers.

Security in MANETs is an active area of research. Most related work in this area covers the integrity of routing tables under the assumption that communication devices are trustworthy [1], [2]. Many recent security threats originate from malware such as viruses, root kits, trojans or targeted software attacks. So called wormholes, black holes, flooding and many other attacks are known to seriously disrupt network services. After compromising a system, such attacks can be easily conducted.

This work was partially supported by the SecFutur EU FP7 project.

Within this paper, concepts from the domain of Trusted Computing (TC) [3] are integrated to provide a solution for the establishment of trust between nodes of mobile ad-hoc networks. The approach supports trusted and secure authentication as well as secure routing and payload transmission between neighbours via established and trustworthy channels. Neighbours are nodes in a MANET that are in range of a single transmission device. In the proposed *MANET of trustworthy members* (TrustMANET) each member is independently able to detect malicious neighbours and exclude those from participation.

We focus on protecting stored keys by cryptographic strong hardware and derive therefrom trustworthy shared secrets at runtime, contrariwise to e.g. common pre-shared key Solutions (WPA2 [4]), the trust relies on a single deployed key and software means. The approach is based on proving the network component's software integrity to each counterpart before establishing a trusted and secured channel for protected transmission in a TrustMANET.

The following section presents background informations of the domain of TC as well as a brief overview of the B.A.T.M.A.N. (*Better Approach To Mobile Ad-hoc Networking*) protocol [5], which is used in this work. A security analysis and derived requirements are provided in Section III. Section IV includes the most important aspects of the definition of a new protocol. Preconditions and other aspects in the context of key management as well as a short example topology and the resulting 'chain of trust' is given. Finally, measurements of a first prototype implementation with the focus of the integrated TPM are shown in Section V. An outlook summarises current and future research activities.

II. Background

A. Trusted Computing

Trusted Computing (TC) as it is described by the TC Group offers a variety of security concepts that are intended to establish a higher level of security amongst *Information and Communication Technology* (ICT) systems. Especially the *Trusted Platform Module* (TPM) [6] can provide a powerful and hard to penetrate root of trust for measurement in MANET nodes. Its architecture, consisting of cryptographic algorithms, persistent registers for keys and hash values as well as I/O interfaces, is similar to commonly known smart card technologies. Implemented as an onboard hardware chip, it serves as a reliable

978-1-4673-5939-9/13/$31.00 ©2013 IEEE

source of trust. *Trusted Computing Systems* (TCS) are equipped with such a TPM, trustworthy booting routines, a trustworthy operating system and trustworthy applications. Starting from the hardware-based TPM, a TCS is able to track all subsequently activated software components in a reliable way that cannot be compromised by software means. In practice, a so called *trust model* must be defined. It depends on requirements of certain use cases and it determines which components (e.g. binaries of boot routines) are subject to measurement and which are not. A TCS preserves hash values of measured components in the *Stored Measurement Log* (SML). TPMs are capable of reporting and signing such a system state (*attestation*) and with the help of appropriate protocols, the attested state can be provided to and verified by remote systems (*remote attestation*). *Attestation Identity Keys* (AIK) can either identify nodes during the attestation process or provide pseudonyms. A more detailed introduction to TC from the perspective of ad-hoc communication can be found in [7].

B. B.A.T.M.A.N. (advanced)

The B.A.T.M.A.N protocol [5] is a proactive routing protocol for multi-hop mesh networks. The motivation of this protocol has its seeds in the shortcomings of other mesh routing protocols and is constantly being further developed by its community. The protocol turns out to be a very promising candidate for a first prototype implementation.

Decentralization of the routing information is implemented by every node keeping track of only the next hop to all targets. B.A.T.M.A.N. provide routing on layer 2 by encapsulation of frames and forwarding them across the mesh. Its kernel module appends frames, transmitted from the IP stack via a virtual bat0 interface, to B.A.T.M.A.N. headers and then sends them out on the physical interface. B.A.T.M.A.N. is typically used in WLAN ad-hoc mode of the physical interfaces and relies on protocol layers below to provide any kind of security. The protocol is proactive in the way that a mesh node constantly announces itself to its neighbours (every second by default) and relies on them to forward this information through the whole network. This results in permanently flooding the network with this informations. Multiple so called Originator Messages (OGM) are aggregated into a single frame, thus mitigating some of the overhead. The more nodes participate in the network the more resources will be used by this proactive approach. However, there is no initial route discovery delay, since routes towards all nodes in the network are known after a certain time. The metric of the B.A.T.M.A.N. protocol is calculated by counting OGM packets[1] received from neighbours (RQ) and of the echo of these OGMs, since an OGM is re-broadcasted and seen by the sender (EQ). Finally, the so called Transmission Quality (TQ) metric can be calculated by $TQ = EQ/RQ$ for a single hop. The complete route metric is calculated hop-by-hop. Additionally, a hop penalty is added, so that the metric favours short over long routes.

[1] within a sliding window of N sequence numbers to calculate the percentage of received OGMs.

III. Security Considerations

We aim at achieving a security enhanced MANET protocol that fulfils the following main objectives. The objectives mitigate threats that are relevant in practice. The proposed concept is described in subsequent chapters.

Protection of communication channels: In MANETs, communication between wireless devices is realized via an open broadcast medium. Compatible devices, in the range of a sending device are capable of receiving all contents of the transmission. Furthermore, they are capable of sending similar or equal contents on the medium. So far, security was not in the focus of the design of these networks. The proposed concept implements mechanisms for the protection of communication channels. It achieves confidentiality of all transmitted data on a hop-by-hop (direct link) basis and it protects from eavesdropping. Authentication and integrity assessment of a remote device precedes any data transmission. Protected communication channels are established in the field. All devices within transmission range exchange shared secrets for the protection of transmissions. The key exchange mechanism also uses the TPM whereby hardware protection against man-in-the-middle -type threats is achieved.

Protection of privacy: The provided solution protects the privacy of users of a device against peers. Unintended traceability, recognition and assignment of single device, and thereby its user, is confined to the link-layer. Pseudonymous TPM keys, the secured key exchange and transmission mechanisms support the protection of privacy. Solutions on higher and lower communication layers, as well as revealing device characteristics, are not in focus of this work. Cross-layer security is e.g. discussed in [8].

Protection of routing tables: Routing tables have to be protected from malicious manipulation in order to counter a variety of threats. Unsecured MANETs suffer from outsider and insider attacks, aiming to inject wrong routing information (e.g. black hole, loops). The dissemination of maliciously manipulated routing information must be prevented. For our solution it is assumed that devices with a correct software state do not manipulate routing information maliciously. Thus, attacks either come from outsider devices or from manipulated software on known devices. The TPM and its integrity measurement mechanisms allow devices to recognize manipulations of neighbouring devices. Routing messages of manipulated devices are dropped and not forwarded in the network.

Protection of cryptographic keys: Capturing of devices by an adversary is a serious concern for mobile equipment. Especially, pre-shared keys need to be protected even if devices are stolen. The provided solution does not require any pre-shared and MANET-wide symmetric keys which are expected to increase the vulnerability of the whole network. Instead it relies on asymmetric keys stored in the TPM. Identity keys and storage keys cannot be compromised by software means. Physical manipulations to TPMs are possible but difficult, expensive and time consuming. All other cryptographic keys utilized in the communication between devices are freshly created, bound to a well known system state and of short temporal validity.

IV. ARCHITECTURE

The TrustMANET protocol establishes protected links between pairs of trustworthy neighbours. Such TrustMANET-Links allow senders to securely transmit typical BATMAN data and routing information whereby common outsider attacks can be prevented by the established cryptographic routines. Nodes achieve a TrustMANET-Link to a neighbour with completion of the following protocol's handshake, checking the counterpart's systems state by the attestation as mentioned before (see II-A).

A. Protocol Design

The handshake defines three processing steps and three types of messages. With completion of the protocol steps, a pair of neighbouring nodes have accomplished:

- Verification of the integrity of the neighbour device
- Authentication of each other
- Exchange of shared secrets for a secure communication

None of the transmitted messages are relayed to the neighbour until completion of the handshake. Table I depicts the required message flow and the establishment of the TrustMANET.
In preparation of the protocol, a set of prerequisites are established beforehand:

- The *take ownership* procedure of the TPM has been performed, which embodies certain credentials in the device [6] and initiates the TPM by locking it from everybody except the owner.
- Each device obtains an asymmetric AIK pair (public and private key) and one validated public key for each operative device. These keys are applied in the operative environment during the TrustMANET handshake as described in the following protocol sequence (see Table I).

1) TCall: Nodes (A and B in step 1) that are not yet connected, frequently and asynchronously broadcast *TrustMANET Calls* (TCalls). Such calls contain a plain text fresh random number, a Diffie-Hellman (DH) public key (S^A_{pub}) and an encrypted routing payload, within the constantly broadcasted Originator Message (OGM) of the BATMAN message format. The routing message remains inaccessible until completion of the handshake, thus not decryptable. Not yet connected nodes simultaneously exchange random numbers within TCall messages. The pair of DH public keys (S^A_{pub},S^B_{pub}) is applied to create a shared secret (S^{AB}) between A and B as it is described by the DH Key Exchange [9]. The DH public key is replaced after a certain time to ensure freshness to any other ongoing authentication processes. As follows by the reattestation, freshness of trust is separated from the established secure channel after the authentication, also to prevent entry points for DoS and Deauthentication Attacks by injecting TCalls with randomly but new S^A keys, easily triggering expensive TPM calculations.

2) TQuote: Step 2 of the handshake can be performed immediately after a random number S_{pub} of an unconnected node has been received (TCall). Both involved nodes (A and B) transmit a Quote and the corresponding SML. Quotes are obtained from the TPM. They include values stored in *Platform Configuration Registers* (PCR) of the TPM. In this work a single PCR, register 10, was used. All Quotes are signed by the AIK. As referred in Section II-A, a counter party can assess the status of an attesting party on basis of the PCR values in comparison with the SML. Depending on its policy, if e.g. only binaries are measured, the old PCR value is concated with the hash value of a binary and afterwards hashed again as well as irrevocable rewritten to the PCR. The SML contains the single hashes to cross-check. To optimize the overhead, a SML can be deployed before (cf. prerequisites) and it is possible to only send the index of the hashed components to shrink the number or size of packets which have to be transmitted. Fake and long SMLs (or indices lists) are not possible to be used for DoS Attacks, since the channel is verified to be valid by the signature of the AIK before and protected by the exchanged secret S^{AB}. Inclusion of a hash, generated and signed by the TPM's AIK, over the counterparts and the own DH-Pub Key (PCR Nonce) protects the communication from Replay and Man-in-the-Middle (MITM) Attacks and guarantees freshness.

3) TData: A TData message is sent if verification of the counterpart's Quote and comparison of the SML succeeded. This message is encrypted by the shared secret S^{AB} and transmits a symmetric key R for broadcast transmission.

To simplify the protocol, the key S^{AB} is used for unicast and R for broadcast transmission. As mentioned, the cryptographic routines can be exchanged. Another type called *TPayload* could take place to separate plain data transmission and key exchange. As result the key deployment got its own key, but an additional unicast key must be deployed that replaces (in this scenario) S^{AB}. Anyway, TData messages might be used to renew keys via the established secure channel. After all, both counterparts are authenticated to each other.

0. Preconditions: Device set-up prior to operations.	
$A:$	AIK^A_{priv}, $AIKCert^B_{issuer}$, $trust\text{-}model$
$B:$	AIK^B_{priv}, $AIKCert^A_{issuer}$, $trust\text{-}model$
1. TCalls: Asynchronous broadcast transmission.	
$A \rightarrow B:$	S^A_{pub}, $enc\{OGM\}_{R^A}$
$B \rightarrow A:$	S^B_{pub}, $enc\{OGM\}_{R^B}$
2. TQuote: Asynchronous unicast transmission.	
$A \rightarrow B:$	$enc\left\{Quote(S^A_{pub}, S^B_{pub}, PCR_{o..n})_{AIK^A_{priv}}, SML^A\right\}_{S^{AB}}$
$B \rightarrow A:$	$enc\left\{Quote(S^B_{pub}, S^A_{pub}, PCR_{o..n})_{AIK^B_{priv}}, SML^B\right\}_{S^{AB}}$
3. TData: Asynchronous unicast transmission.	
$A \rightarrow B:$	$enc\left\{data := \{payload \,||\, R^A\}\right\}_{S^{AB}}$
$B \rightarrow A:$	$enc\left\{data := \{payload \,||\, R^B\}\right\}_{S^{AB}}$

TABLE I
TRUSTMANET AUTHENTICATION

4) Re-Attestation: Triggering the protocol periodically between neighbours of the already established TrustMANET (interval depends on the use case) is needed, so that, by verifying the TQuotes, freshness of the TrustMANET is guaranteed. Renewing shared secrets between already connected devices is also controlled by the reattestation. This results in renewing the DH public key, so generate a fresh unicast key, which

is permanently accepted after receiving TData. As described before, TData, as a reply to a valid TQuote messages, contain the new broadcast key. If any verification fails, a node is excluded by the verifier and all keys of the classified malicious node are discarded. The nodes own broadcast key also has to be renewed after a reattestation failed. Detailed concepts are needed to improve the efficiency and to avoid leaking informations to the detected malicious node, while deploying the new broadcast key to the TrustMANET, thus minimizing resulting unidirectional connection interrupts. In this work, the verifier stops its broadcast traffic until the TrustMANET is freshly attested again.

B. Overhead

The packets of the introduced protocol messages (TCall, TQuote, TData) are transmitted using Ethernet, in the same way the B.A.T.M.A.N. protocol does. Figure 1 shows a modified OGM packet extended as a TCall. Besides a common and unchanged Ethernet Header with destination and source MAC as well as the protocol type (*Typ*, which was changed to an unassigned one), only one byte was used for a packet type (*P*) to identify the different B.A.T.M.A.N. packets. Following, is the DH public key and the encrypted OGM payload. The overhead consequently depends on the size of the DH public key and the encryption which is chosen for, as well as on the size of, the payload. The random numbers and their size have to be chosen according to the recommendations for the DH protocol [9], [10].

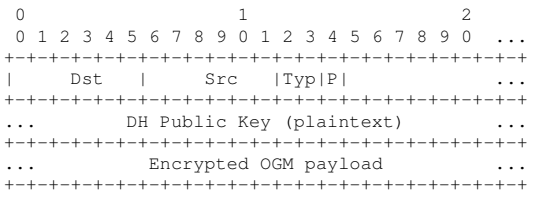

Fig. 1. Extended OGM packet

All packets are designed in the following way:

- Ethernet Header (*Dst*, *Src*, (Protocol) *Type*)
- Packet Type (*P*)
- Plain text (like DH public key or IV if needed)
- Encrypted Payload

TQuote packets use 316 bytes for the Quote which consists of two PCR Nonces of each 20 byte, the PCR register of 20 bytes and the AIK signature of 256 bytes. The size of the SML depends on the trust-model and what was measured. An approach of transmitting only a list of indices and deploying a lookup of the trust-model by the preconditions was already introduced. The size of TData packets depends on the payload and the chosen encryption for transmission, what is true for all kind of packets in the TrustMANET.

C. Key Management and Chain of Trust

Each node of the TrustMANET maintains its own key table (such as Table II), containing the deployed keys. Nodes are addressed using their unique (but insecure) *Media-Access-Control* (MAC) addresses, but securely identified using the AIKs. If a trustworthy connection to a neighbour could be established, the resulting shared secrets (S- and R-Key) are added to the table. Furthermore, each entry holds a public key (TPM-Key), which is required during the attestation process.

Column *Best Next Hop* of Table II contains the best next hop from node D to a destination MAC_X for the example topology of Figure 2. Note that the actual best next hop to reach node A, what is either B or C, is chosen by the underlying B.A.T.M.A.N. protocol. Because node A was not yet in reception range of D, no S- and R-keys were exchanged, but it may communicate with D via B or C. If A is entering the reception area of D, a handshake is automatically performed to establish a direct connection. Concepts for deploying keys and handing trust of neighbours over two nodes, which are not in direct reception range, might be integrated in further concepts and research.

MAC	TPM-Key	S-Key	R-Key	Best Next Hop
MAC_A	AIK^A_{pub}	n.n	n.n	either B or C
MAC_B	AIK^B_{pub}	S^{BD}	R^B	B
MAC_C	AIK^C_{pub}	S^{CD}	R^C	C
MAC_E	AIK^E_{pub}	S^{DE}	R^E	E

TABLE II
ROUTING AND KEY MANAGEMENT OF NODE D

Figure 2 illustrates a short example topology and shows the *chain of trust*, which is formed by forwarding (2, 3) an OGM of node A (1). Node A is directly connected to B and C, as well as D to B, C and E via TrustMANET-Links. The malicious node M is excluded from participation. Node D, which is in the transmission range of M, will not relay any of its messages (4) until M provides adequate proof of integrity. Finally the messages of A reaches E by the forwarding and trusted node D, delivering the OGM of A by prior receiving it either from B or C, so that a secure communication is established by a hop-by-hop trusted chain. (E is a designated node serving as gateway to other networks, cf. Conclusion, VI.)

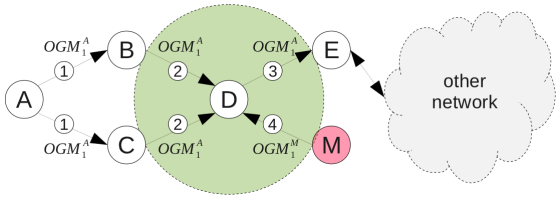

Fig. 2. Example Topology - Chain of Trust (A to E)

V. MEASUREMENT

The proof of concept implementation of the introduced protocol can be separated into two components. The first component (TMModule) is directly implemented as part of the B.A.T.M.A.N. kernel module and handles the low level frame communication. The second component (TMDaemon)

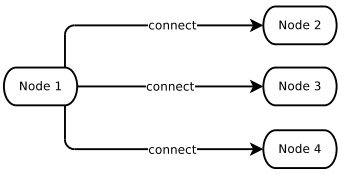

Fig. 3. Experimental Network Overview

is realized in JAVA as a daemon running in user space and is responsible for assembling and analyzing the messages of the authentication process. This also involves operations which are executed on the TPM. The interprocess communication of the components is handled by a character device.

The measurements presented in this work focus on the authentication process and in particular on the operations executed by the TPM, as these were the most time consuming operations. The measurement of the implementation was taken on a *Genuine Intel(R) CPU N270, 1,60GHz* with *2GB* of *RAM* with the Linux kernel version *3.0.0-14 (32-bit OS)* and an *Infineon TPM 1.2 4bit GPIO*. The experimental environment consists of four nodes (Node 1 to 4), as shown in Figure 3, which are physically connected over a 100MBit network switch[2]. Node 1 acts as the supplicant, which tries to establish a trustworthy connection to the remaining three nodes (Node 2-4). When the authentication process for two involved nodes is finished and thus the trustworthiness of one another is verified, the connection is established. Figure 4 depicts the measurement of Node 1 connecting to Node 2-4.

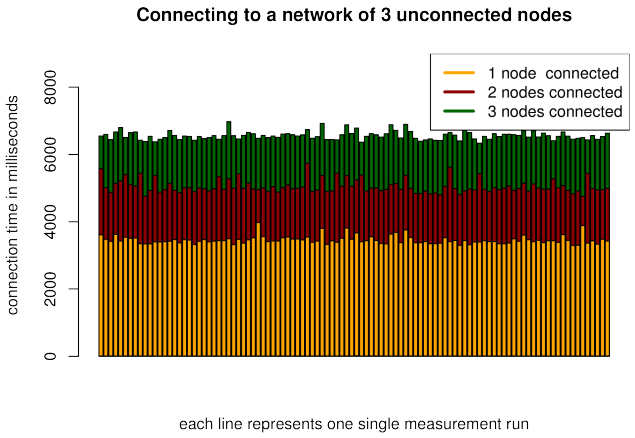

Fig. 4. Supplicant joining a network of 3 nodes.

The first connection between Node 1 and one particular counter-node is established after 3457 ms in average. The second connection is established after 5049 ms and the third after 6566 ms in average. As this measurement includes TPM operations, which are considered in general as slow, Figure 5 shows a measurement of the TPM Quote operation.

[2]A wired Ethernet connection, here compatible to the B.A.T.M.A.N. protocol, was preferred to eliminate possible wireless interference, since measurements are focused on the authentication process in the context of the TPM calculations.

The generation of a single Quote takes 1604 ms in average. As each successful connection involves a Quote generation on both involved nodes and the supplicant starts the Quote Operation after receiving the counterpart's valid Quote, the Quote generation takes 3208 ms in sum. Compared to 3457 ms, consumed by the establishment of the first connection, approximately 93% of the total time is taken only to generate the Quotes by the initial authentication process.

Additionally the TPM Quote operation may only be executed exclusively on the TPM. This means in order to actually establish a connection to three other nodes, the operation is executed sequentially at the supplicant. For the experiment this means that the Quote-generation for the second connection attempt blocks until the first Quote is computed. This behaviour explains the connection times shown before, as approximately 1600 ms lie between each different connection establishment. Any other measurement, which was taken in the experiment, becomes negligible if compared to the Quote-generation process. In the current protocol specification it is not possible to generate the Quote once on the supplicant, and reuse the Quote for multiple other nodes, as it contains input values unique to the corresponding counter-node.

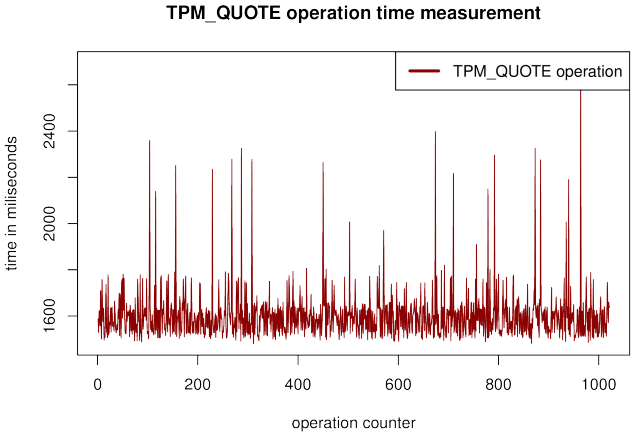

Fig. 5. Generating and signing the TPM Quote.

The experiment shows that the TPM Quote generation is the current bottleneck of the authentication process. Further functions (packet encryption, Quote verification, message assembling and analysing, network delay) causes the rest of the consumed overall time, which is approximately 200 ms. The optimization of those functions, but especially the optimization of the time expensive TPM Quote generation is the focus for further research and needs to be solved before real world implementations become practical.

VI. Conclusion and Outlook

The proposed protocol establishes trust relationships on the link layer of a mesh network. Concepts from the domain of Trusted Computing are integrated. The Trusted Platform Module serves as a root of trust, that enables devices to mutually

attest system states and to establish protected channels for routing and payload transmission. This design achieves the stated security requirements.

Especially the verification of devices' states and the assumed transitivity of trust in the MANET is a subject of future work. A specific list of reference values is derived from real world application environments. Devices may attest installed software packages, executable binaries, memory areas, cryptographic keys, routing tables or others. Systems appraising an attestation need to be provided with all necessary information (e.g. policies or white lists of expected measurements). If such information can not be stored in advance, additional concepts are needed.

This solution shows approaches to separate attacks with the aim of either compromising a system or injecting falsified data from the outside. It may exclude insider threats in general. For example DoS threats might be only associated with outsider attacks. Any insider node, executing a DoS, should be identified if the reference value list is defined clearly, thus the malicious DoS software would fail attestation. DoS from the outside, that target at the TPM calculations, is a topic for future research.

However, as a counter example, if a trust model defines virtualisation as a trusted task, but nothing further, a virtual machine (VM) is legally launched, and could inject falsified data, yet attestation of the host system indicates no issue. This shows how thoroughly a trust model must be defined and the analysis of use cases must pay attention to such issues.

Furthermore, interoperability to other network standards, such as used by common LAN backbones, i.e. 802.3 Ethernet, is of importance. It needs to be considered how TrustMANET devices can interact with networks of different layer technologies, building a heterogeneous seamless communication system. Gateways are already part of the TrustMANET, however there has been no evaluation of how a trustworthy connection can be maintained over another layer 2 protocol. In such scenarios, a gateway should not be implemented as the single point of failure. Not only other operative scenarios must be extended but also the protocol design and its functionality must be optimized. Therefore, a pure ANSI C implementation opens up the possibility to optimize the protocol. Moreover, the TPM Quote calculation is stated to be around 500 milliseconds as given by the vendor, but on the hardware used, longer times were measured. Thus, these times may vary on different TPM chips as well as the corresponding software interface calls (e.g. Java or C). This is to be evaluated in the upcoming research, but it additionally points out the expensive calculations, showing the bottleneck of the protocol, which is related to the TPM chip. A more generic solution, which does not require modifying the OGM packets, is planned. Additional approaches where neighbour-discovery and attestation are separated, might offer better characteristics in real world scenarios. After that, empirical studies should reveal important characteristics, such as *performance* or *throughput* in several wireless scenarios. Especially the TPM must be addressed to improve the performance of authentication and attestation. Interesting approaches are presented in [11].

It needs to be questioned whether additional concepts are needed to achieve a network that is scalable and capable to deliver the intended services. The presented first measurements prove to be promising as the protocol is stable and the concept tackles several security issues of today's MANETs.

REFERENCES

[1] M. Carvalho, "Security in mobile ad hoc networks," *Security & Privacy, IEEE*, vol. 6, no. 2, pp. 72–75, 2008.

[2] N. Garg and R. Mahapatra, "Manet security issues," *IJCSNS*, vol. 9, no. 8, p. 241, 2009.

[3] C. Mitchell, *Trusted computing*. Iet, 2005, vol. 6.

[4] IEEE, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements (IEEE Std 802.11i-2004)*, Institute of Electrical and Electronics Engineers, Inc., July 2004.

[5] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking (B.A.T.M.A.N.)," 2008, http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00.

[6] *TPM 1.2 Main Specification*, Trusted Computing Group, 2011, http://www.trustedcomputinggroup.org/resources/tpm_main_specification.

[7] N. Kuntze, A. Fuchs, and C. Rudolph, *Lecture Notes in Computer Science 6033*. Springer, 2010, ch. Trust in Peer-to-Peer Content Distribution Protocols, pp. 76–89.

[8] D. Kidston, L. Li, H. Tang, and P. Mason, "Mitigating security threats in tactical networks," in *IST Panel Symposium, Military Communication and Networks*, 2010, Wroclaw, Poland.

[9] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[10] D. V. Ramarathnam, "Significant bits of secret keys in dh and related," *Advances in Cryptology -CRYPTO '96*, 1996.

[11] F. Stumpf, A. Fuchs, S. Katzenbeisser, and C. Eckert, "Improving the scalability of platform attestation," in *Proceedings of the Third ACM Workshop on Scalable Trusted Computing (ACM STC'8)*. Fairfax, USA: ACM Press, October 31 2008, pp. 1–10.

[12] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004, pp. 132–145.

[13] N. Kuntze, A. Fuchs, and C. Rudolph, "Reliable identities using off-the-shelf hardware security in manets," in *Proceedings of the International Symposium on Trusted Computing and Communications (TrustCom 2009)*, 2009, http://sit.sit.fraunhofer.de/smv/publications/download/TrustCom09.pdf.

[14] H. Tang and M. Salmanian, "Lightweight integrated authentication for tactical manets," in *ICYCS*, 2008, pp. 2266–2271.

[15] M. Jarret and P. Ward, "Trusted computing for protecting ad-hoc routing," in *Proceedings of the 4th annual Communication Networks and Services Research Conference (CNSR'06)-Volume 00*. IEEE Computer Society Washington, DC, USA, 2006, pp. 61–68.

[16] T. C. Group, 2011, http://www.trustedcomputinggroup.org.

[17] SourceForge.net, "Integrity measurement architecture (ima)." http://linux-ima.sourceforge.net/.

[18] M. Ikeda, E. Kulla, M. Hiyama, L. Barolli, and M. Takizawa, "Experimental results of a manet testbed in indoor stairs environment," in *2011 IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2011, pp. 779–786.

[19] S. Balfe, A. Lakhani, K. Paterson, and et al., "Trusted computing: Providing security for peer-to-peer networks," in *Proceedings of the Fifth International Conference on Peer-to-Peer Computing (P2P05)*. 2005, IEEE, Konstanz, Germany, 2005, pp. 117–124.

[20] by the Institute for Applied Information Processing & Communication (IAIK), "Trusted computing for the java(tm) platform," 2011, http://trustedjava.sourceforge.net.

[21] David Johnson, Ntsibane Ntlatlapa, Corinne Aichele, "A Simple Pragmatic Approach to Mesh Routing Using BATMAN," Meraka Institute, CSIR, Tech. Rep., 2008.

[22] A. W. Kwan-Wu Chin, John Judge and R. Kermode, "Implementation Experience with MANET Routing Protocols," Sydney Networks and Communications Lab, Motorola Australia Research Centre, Tech. Rep., May 2003.