Self-sovereign Identity Management in Wireless Ad Hoc Mesh Networks

1st Michael Grabatin
Institute for Software Technology
Universität der Bundeswehr München
Neubiberg, Germany
michael.grabatin@unibw.de

2nd Wolfgang Hommel
Institute for Software Technology
Universität der Bundeswehr München
Neubiberg, Germany
wolfgang.hommel@unibw.de

***Abstract*—Verifying the identity of nodes within a wireless ad hoc mesh network and the authenticity of their messages in sufficiently secure, yet power-efficient ways is a long-standing challenge. This paper shows how the more recent concepts of self-sovereign identity management can be applied to Internet-of-Things mesh networks, using LoRaWAN as an example and applying Sovrin's decentralized identifiers and verifiable credentials in combination with Schnorr signatures for securing the communication with a focus on simplex and broadcast connections. Besides the concept and system architecture, the paper discusses an ESP32-based implementation using SX1276/SX1278 LoRa chips, adaptations made to the lmic- and MbedTLS-based software stack, and practically evaluates performance aspects in terms of data overhead, time-on-air impact, and power consumption.**

***Index Terms*—Self-sovereign Identity Management (SSI), MANET, Internet of Things (IoT), Identity Management, LoRa**

I. Introduction

Wireless ad hoc nets (WANETs) can be used to build wireless networks in regions where other network infrastructure is lacking or not viable. If the nodes of the network are mostly mobile, the network is called mobile ad hoc network (MANET). Mesh networks can be distinguished by used technology, organizational structure, or system permanence. Example MANETs include:

- Guifi [1]: Wi-Fi-based, static community network
- GoTenna [2]: commercial ad hoc network for individuals
- Meshtastic [3]: LoRa-based, ad hoc network

The definition of MANETs requires routing among the nodes. There are also similar radio networks that mainly consist of entities broadcasting messages relevant to other entities within reception range. Because those networks lack the usual forwarding of messages for other entities within the network, but face similar problems for secure identification and authentication of other peers and their messages, they are referred to this paper as *mesh-lite* networks. Examples of those networks are:

This work is partly funded by the Bavarian Ministry for Digital Affairs (Project DISKURS/STMD-B1-4140-1-4). The authors alone are responsible for the content of the paper.

- Automated Identification System (AIS) [4]: commercial broadcast network to assist with nautical navigation as well as search and rescue operations
- ADS-B [5]: commercial broadcast network to assist with navigation in aviation

Most research for MANETs focuses on nodes' Identity and Access Management (IAM) either based on certificates or reputation. By choosing one or the other, the systems' general-purpose usefulness is limited, as certificate-based approaches lend themselves to more structured deployments while unstructured networks benefit from reputation-based systems.

Self-sovereign Identity Management (SSI) offers reliable, decentralized, and secure identities and authentication. This allows for both approaches without having to settle on one or the other. The verifiable credentials (VCs) can be used as both, a substitute certificate or reputation token.

This paper develops and evaluates an approach of integrating SSI in MANETs while maintaining compatibility with low-powered devices (i. e., less than 512kB RAM and less than 10MB storage) and bandwidth-limited communication links (i. e., less than 256 bytes per message).

A. Contribution

In this paper we show an application-agnostic integration of SSI in mesh and mesh-lite environments. Especially low powered Internet-of-Things (IoT) networks, which may only transmit and cannot constantly receive messages, cannot properly utilize protocols that rely on bilateral message exchange to prove an identity or associated attributes, which is why they are avoided in this approach. Further contributions are:

- Identification of nodes within the mesh network via Decentralized Identifiers (DIDs).
- Signing extremely short messages using suitable Schnorr signatures.

To accommodate for low-powered IoT devices and to prevent congestion of radio links a key aspect of the proposed solution is to reduce the amount of data overhead that has to be transmitted.

B. Structure

The rest of the paper is structured as follows: Section II discusses related work and similar approaches. Section III

elaborates on the problem and presents more background information. The proposed solution is presented in Section IV and an experimental implementation is described in Section V. The implementation is evaluated in Section VI and final thoughts and development ideas are outlined in Section VII.

II. Related Work

The security of common IoT protocols, including LoRa and LoRaWAN, has been analysed by multiple researchers. For example [6], [7] both show that LoRa features confidentiality by encryption and can protect message integrity and supply end-point authentication by using hashed message authentication codes (HMACs). Using HMACs as an integrity and authentication mechanism only works in scenarios where there is a pre-shared key (PSK) for every pair of communicating nodes. This is sufficient for the LoRaWAN star topology but its usefulness in a distributed mesh network is limited.

One method for authenticating nodes and to determine admission to the network is described by [8], which makes use of threshold cryptography. The authors argue that a centralized decision point would hurt the MANET's performance, as it would be unreliable, especially in short-lived and constantly moving networks. A centralized approach would also introduce a highly worthwhile target for adversaries. The threshold approach enables a group of t out of n nodes to grant access to the network. The presented bivariate polynomial secret-sharing protocol is non-interactive, secure, and efficient solution which is mindful of power consumption, computational, and communication limits of mobile devices. In most scenarios of short-lived MANETs the strengths of this approach do not pay off, as for the nodes to know which other nodes to admit to the network they need prior information about their identity. In that case a (temporary) public-private-key infrastructure can achieve a more robust and efficient access control.

Another method for identifying and authentication nodes is described by [9]. The authors describe a method for creating a Pretty Good Privacy (PGP)-like trust network for MANETs using self-certifying identity-based cryptography. In identity-based cryptography a node's public key can be derived from publicly available information (e. g., the name) of the node. The corresponding private key is calculated by a private key generator with a secret parameter obtained from the system authority. A network of trusted nodes is built which is efficient because it does not rely on certificate chaining, like regular PGP does. Using this schema and two different trust metrics, nodes can create a trust network without involving a trusted third party (TTP) or any other external information. Like with PGP a problem of this approach is again the lack of information of a node on whom to trust, even though the authors describe how to calculate transitive trust.

As our planned integration of SSI with MANETs also strongly connects to Blockchain and Distributed Ledger Technology (DLT) research, related work from those fields is also considered. The authors of [10] describe an extension to the open-source off-grid communication device MAZI [11]. This extension utilizes a private permissioned Blockchain based on Hyperledger Fabric to facilitate a secure and distributed identity management and data sharing platform. Further intended applications include voting, chat, file sharing, IoT, and Service Level Agreement (SLA) monitoring.

Another project by [12] shows an extension for the WANET Guifi, which is primarily deployed in Spain with over 36k nodes and almost 70k kilometres of total links [13]. They also introduce a private permissioned Blockchain to the network, but are focused on providing an automated accounting system that incentivises providing resources to other participants.

III. Background

At first the possible attacks on MANETs are described in Section III-A. Further requirements and key parameters necessary for a solution are described in Section III-B, which also elaborates on the targeted wireless physical layer protocol LoRa in more detail. The mathematical considerations for a solution using Schnorr signatures are described in Section III-C. Section III-D shows the requirements and benefits of using SSI.

A. Attack Vectors

Many open mesh and mesh-lite radio networks suffer from the same problems, which arise from the underlying concepts and designs. A central authority, which could effectively allow and deny entities from joining the network, is missing and the network is designed to be open and accessible to many devices. No encryption, at least for the management part of the network, enables observers to gain information about participating entities. This can allow entities to spoof other entities' identities, something that can potentially be used to forge messages or facilitate replay attacks. An attacker in this kind of network may also gain advantage by creating many identities that do not correlate to actual real-world entities, a technique also known as Sybil attack. This would be useful to work around rate-limiting by other network participants or to sway opinion in any group consensus or web-of-trust scenario.

With a more advanced attack method the attacker can hijack connections of other nodes and manipulate the data contained in their messages in real-time. This prevents others from detecting modification as no messages with potentially conflicting contents arrive.

Both impersonation and hijacking can be used to attack susceptible resource management protocols in order to exhaust system resources, which can result in Denial-of-Service (DoS) of all or selected entities. For example, AIS class A transmitters are susceptible to this as they send in fixed time slots and attackers may block or reserve some (or all) time slots without proper authentication [14].

Last but not least, wireless transmissions are prone to jamming, by having a transmitter block the usable frequencies, which also results in a DoS attack, at least in the vicinity of the jammer. IAM on the Internet is usually done at the application level. For MANETs (e. g., except Guifi, all the examples mentioned above) the separation of layers is not always that strict, resulting in device and user identities being used as

addresses for routing as well as IAM. As identifiers need to be announced to all other nodes, it is especially difficult to design privacy-preserving identifiers in such a scenario. The preferred solution of pairwise identifiers does not work for broadcast messages or mesh-routing either.

B. LoRa

LoRa is a low-power-optimized physical layer protocol for wide-area networks operating in license-free frequency bands (Europe 868 MHz, North America and Australia 915 MHz, and Asia 923 MHz). The range of common LoRa transmissions is about 5 km to 10 km (with direct line of sight even significantly longer). It is commonly used for hub and spoke LoRaWAN networks, but can also be used independently for other applications, e. g., direct or mesh communication. Both (LoRa and LoRaWAN) standards are developed by Semtech and the LoRa Alliance.

LoRa specifies a physical communication protocol based on proprietary chirp spread spectrum modulation. The amount of spreading used can be configured by the application and affects transmission time and range. Higher spreading factors increase the time on air but also increase range and reliability. The structure of a LoRa packet consists of a preamble, followed by a header and header checksum, the payload and payload checksum [15]. The header contains the length of the payload as a byte-sized field. As a result the maximum payload length per packet is 255 bytes. One of the main challenges of signing LoRa packets is to use efficiently the limited space such a packet provides. Especially, in order to counter the attacks described in Section III-A, an efficient signature method is required to protect the packets' integrity and to authenticate the sender.

C. Schnorr Signatures

Schnorr signatures [16] provide a method to construct smaller signatures while being available for some time and having been proven secure by multiple researchers [17], [18]. Combining the original Schnorr signatures with elliptic curve cryptography [19] can achieve even smaller signature sizes.

When using a cyclic group of prime order p with a size of 256 bit, the two signature values s and R can be stored in 32 bytes and 64 bytes, resulting in a combined signature size of 96 bytes. The sizes can be further improved by selecting different hash functions and curve parameters as well as using compression, which will be shown in Section V.

D. SSI

Finding a way to produce sufficiently short signatures is only one part of the problem of bringing SSI to a LoRa-based MANET. The solution is based on the DID-based SSI system developed by Sovrin [20], [21] and Hyperledger Indy [22], which, more recently, resulted in the Hyperledger Aries project and a complete communication protocol between entities called DID communication (DIDCom) [23].

The basic premise of those SSI projects is to create a DLT that can be used by anybody to register public entities, including their metadata, such as communication endpoints and public keys. The ledger can also be used to define common schemes for expressing credentials in order to have parties agree on some of those schemes for exchanging their users' credentials. Both of those primary functions of the ledger are not essential to possess and use a SSI identity, but they provide a way to define trust anchors within the system. The actual IAM is done through agents, which for consumers might be a smartphone application, whereas for businesses special IAM components are added to their services. An example of how a user would authenticate to an authority and service is depicted in Figure 1. The exchange of a VC through the user's wallet is also shown.

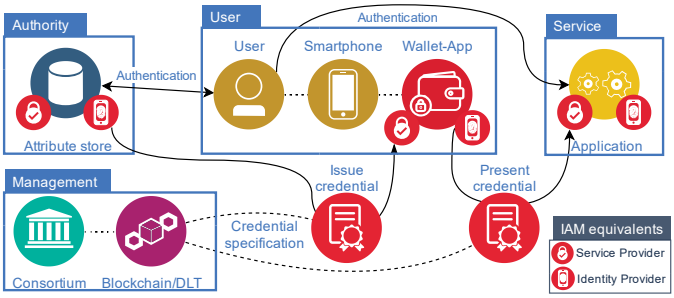

Fig. 1. Example of an SSI exchange

Attributes are attested to entities by the way of VCs. VCs follow credential definitions, which describe the structure and content of a credential. Those definitions can be created, stored, and used on the ledger by any participant. Effectively, they represent more flexible and lightweight certificates. The credentials are created and signed by the issuing entity and passed to the holder of the credential. To use the attested attribute the holder can present the credential to any other entity without involving the original issuer. The verifier can then check the signature of the issuer and – presuming the verifier trusts the issuer – authenticate the holder. This step can be performed completely offline, which makes it particularly interesting for IoT and MANET applications.

IV. System Architecture

As described in Section III-D, the key ingredients to SSI are the DID identifiers and the corresponding DID documents describing the identities. That system is adapted to support offline mesh networks without the need for constant Internet connection and a focus on reducing resources for low-powered devices and low-bandwidth communication methods. At first the focus is kept on device (or radio) identifiers that are used by the network to facilitate identification of nodes and possibly message routing. On top of this there can be an application layer that supports identifying and authenticating users or services. How retrieving DID documents can work in a MANET is described in Section IV-B. Using the contained information to identify and authenticate a node is shown in Section IV-A. Section IV-C details the operations a SSI-enabled MANET node needs to implement.

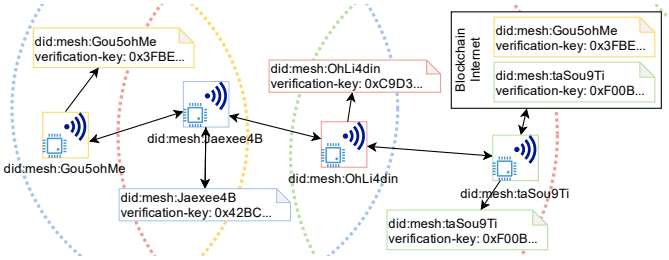

Fig. 2. SSI architecture in a MANET

The resulting LoRa SSI MANET is depicted in Figure 2. It shows the different entities and their respective communication range and direct connections. Each node either publishes its DID document in a ledger on the Internet or within its broadcast domain. In the first case, this allows nodes to learn about other nodes before meeting them. If that is not possible (or intended), nodes use a trust on first use (TOFU) approach to accepting unknown nodes' information.

A. Identification and Authentication

Identification of nodes within the MANET is done via a node ID. The node ID can be generated by taking the first bytes of the hash of the node's public key. The length of the node ID must be chosen to prevent collisions between node IDs while not wasting precious message space, as the sender's and receiver's IDs need to be sent with every message to enable routing the message. The chance of a collision depends on the number of nodes within the network. The likelihood of a collision occurring, when the nodes' IDs are chosen randomly, can be approximated by the formula: $1 - e^{\frac{-k(k-1)}{2N}}$; where k is the number of nodes and N is the number of different possible node IDs [24]. Using 4 byte IDs with about 1000 nodes yields a collision probability of 0.01%. This seems acceptable for small deployments, larger deployments must adapt accordingly.

Identified nodes are authenticated according to the authentication method they specified in their DID document. For our use-case, this defaults to an authentication via a Schnorr signature. In most cases the node's DID needs to be public, i. e., it must be actively announced. This is necessary to identify the "important" nodes, on the one hand, and to route messages through the mesh network, on the other hand. Private DIDs may only be used in direct communication between two nodes.

B. DID Document Retrieval

Within a mesh network nodes may have differing tasks and administrative privileges, e. g., in regard to monitoring and assigning transmission slots to other nodes. Nodes with higher privileges are therefore more important to correctly identify and authenticate than those with lower privileges. Correct identification requires prior knowledge about those nodes by all other participants.

As a result, node identifiers are communicated in a hybrid approach. Identifiers of important nodes are stored on a ledger and need to be retrieved. Which kind of nodes are important is up to the application to decide. The DLT that stores all publicly available entities will contain many identities that are not relevant to a particular application. Association with a specific group of identities can be done by utilizing VCs of one or more authorities in that specific field.

Identifiers of regular, non-public nodes are communicated within the network and are cached by participants when they first appear. The DID document of those nodes must be regularly communicated to all neighbouring and interested nodes.

C. Operations

The following necessary operations need to be implemented to run SSI on a mesh network. The operations are gathered from the previous two sections and SSI methods, as they are outlined for Sovrin's identity ledger by [20].

Register public identity: Public identities must be registered by uploading the entity's DID document in the distributed ledger. **Generate private/pairwise identity:** For privacy-preserving exchanges between two directly communicating entities, a pairwise identity is generated. **Authenticate a node:** To verify the identity of another node, the verification method described in the node's DID document must be used. **Establish verifiable credentials:** Credentials attest specific attributes to an entity. They can either be generated by the entity itself or by another entity. **Store VCs:** Credentials are managed by the individual credential holders and can potentially contain private information. **Present VCs:** In order to use credentials they must be presented to other parties who can then check their validity and contents. **Revoke VCs:** If the assertions made in a credential are no longer valid, the issuing party must be able to revoke the credential. VCs solve it by either actively checking with the VC's issuer or by utilizing zero-knowledge proofs [25]. **Delete identity:** Identities that are no longer needed must be deleted securely to prevent future misuse. Deleting an identity effectively amounts to deleting the DID's corresponding private key.

V. Implementation

The implementation is done for ESP32 boards utilizing a SX1276/SX1278 LoRa chip. The boards commonly feature a dual-core CPU that runs at around 240 MHz and has access to 512 kB RAM. Such boards are available relatively cheap, about 20 to 30 Euros. Common boards are produced by Heltec and LilyGO/TTGO. Most boards feature not only the LoRa chip but also contain Wi-Fi and Bluetooth chips that can be used to link the boards to a smartphone.

To program the ESP32, the Espressif IoT Development Framework (ESP-IDF) is used. The LoRa communication is done via an Arduino `lmic` LoRa library [26], which was adapted to the ESP-IDF. Some changes needed to be made to the `lmic` library to send and receive payloads of more than 64 bytes. This limit is used by many LoRaWAN applications and ignoring longer messages while receiving can conserve energy.

The Schnorr signatures are created following an implementation [27] of said signatures in C using the OpenSSL library for elliptic curve calculations. This implementation was rewritten to utilize MbedTLS instead of OpenSSL, as it is deployed with the FreeRTOS operating system of ESP-IDF.

The network structure and networking protocol is kept simple for this implementation. All nodes listen continuously for packets and repeat them if they haven't already repeated them previously, thus avoiding loops. A time-to-live field in each packet limits the number of hops a packet can travel. More efficient and resilient algorithms could be implemented to improve the transmission effectiveness, but routing algorithms are not within the scope of this research.

A. Packet Format

The exchanged message's structure is defined using Protocol Buffers (protobuf), as shown in Listing 1. Protobuf definitions allow for a implementation independent definition of the used packet format. They are translated to the implementation's programming language using a compiler. For this paper *nanopb* [28] is used to compile the minimal *C* code necessary.

```
message Packet {
  required fixed32 id = 1;
  optional Did from = 2;
  optional Did to = 3;
  required int32 ttl = 4;
  required PacketType type = 5;
  optional bytes payload = 6;
  required Signature signature = 7;
  required bool wantAck = 8;
}
```

Listing 1. Protobuf definition of a message

The `from` field may be excluded if the message is sent using a pairwise DID, as the recipient knows the corresponding DID. Omitting the `to` field signals a broadcast message, not intended for one specific node. Some messages may not have a `payload`, as they are merely presence announcements or – if required – acknowledgements.

The remaining required fields are used as follows: The `id` is an arbitrarily chosen value that is used to prevent routing messages in loops and to protect against replay attacks. If the node can determine the current time, a timestamp can be used as `id`. Packets are propagated within the mesh network, while each node decrements the `ttl` counter by one, until the `ttl` reaches zero. While it is defined as `int32` in the protobuf, the `int32` field uses variable length encoding and a `ttl` between 0 and 255 can always be stored in one byte. The `type` of the packet differentiates between `ping` (general availability broadcast), `ack` (acknowledgment of a message with identical `id`), `disco` (node discovery request), and `msg` (any directed message). The `signature` contains the values s and R of the Schnorr signature calculated over the fields `id`, `from`, `to`, `type`, `payload`, `wantAck` (if they are present). For particularly important messages, the `wantAck` flag can be set in order to request an acknowledgement of a received message. It can only be used in combination with messages of type `ping` and `msg`. Because of the limited communication bandwidth this should be used sparingly.

B. SSI Operations

To test the suitability of Schnorr signatures as a method of verifying the identity of MANET nodes, the two main operations signing and verifying have been implemented.

The signature is generated by computing a RIPEMD-160 hash [29] over all static parts of the packet: `id`, `from`, `to`, `type`, `payload`, `wantAck`. This only excludes the time-to-live, which is changed on each node, and – naturally – the signature field. Here, the RIPEMD-160 hash algorithm is chosen because it can be calculated more efficiently than the 12 bytes longer SHA 2 [30] and – contrary to the equally 20 bytes long but considered broken SHA-1 [31] – there are no known attacks against full RIPEMD-160 [32].

The resulting hash is then signed using the node's private key using the Schnorr signature algorithm. The Schnorr algorithm is used as described in Section III-C in the common configuration with SHA-256 as the hashing algorithm and the elliptic curve *secp256k1* [33]. Changing the hash algorithm and curve parameters to reduce the signature size and increase performance has been evaluated as shown in Section VI-B.

The verification of a packets signature runs through the same steps of hashing the received packet's fields, and verifying the parameters passed in the packet's signature field can be used to check the hash's authenticity.

VI. Evaluation

The implementation described in Section V was used to transmit messages between multiple devices. The experiments where primarily done using two identical Heltec ESP32 Wi-Fi LoRa V1 boards.

A. Security

Evaluation of the system's security is done by showing that the used cryptological primitives are widely considered to be secure. It is not shown or proven that the implementation built here is without faults that may limit actual security.

In Section III-A the following attack vectors for MANETs where shown. The first two of them can be prevented by utilizing the implemented SSI setup.

- Identity spoofing: Due to the use of signatures an attacker can no longer forge messages. The use of message IDs and message counters also prevents replay attacks.
- Connection hijacking: In-flight modification of transmitted messages is immediately detected by the recipient.
- Resource exhaustion: Any attacks that rely on impersonating an authority are prevented by using signatures and having authoritative entities supply the necessary VCs.

B. Performance

The performance of the solution is primarily measured in the overhead produced by adding signatures to the transmitted messages. This way it can be compared against other system parameters (i. e., hash functions and elliptic curves), versions of the implementation (i. e., with specific optimizations), and other approaches altogether. The overhead can be measured for the transmission length (in bytes and time-on-air), computation

steps/time required, and power usage. The first measure can be expressed exactly, while the other two are increasingly difficult to measure precisely. The overhead resulting from the signature can be derived from the Schnorr signature itself. As described in Section III-C, the signature consists of the tuple $\sigma := (s, R)$, where s is a large number and R represents a point on the curve. As the signature algorithm operates in a group $\mathbb{Z}_p$, the largest possible number is determined by p.

Using a p with n bits limits s to n bits. For the *secp256k1* curve this results in 65 bytes needed to represent the point R in an uncompressed [34] form.

The time-on-air is directly correlated to the size of the message and LoRa parameters. Times within this paper have been calculated using https://www.loratools.nl/#/airtime for the commonly used spreading factor SF7, bandwidth 125MHz, and coding rate 1 (4 bits encoded as 5 bits).

The resulting sizes of the signature and the time it takes to transmit it are shown in Table I. As an alternative smaller elliptic curve the parameters for *secp192k1* are also considered. Other, especially larger curves are not considered and yet smaller curves are deprecated for security reasons.

TABLE I
SIGNATURE SIZE AND TIME-ON-AIR DEPENDING ON PARAMETERS

elliptic curve	size (bytes)	time-on-air (ms)
secp256k1	97	~ 160
secp192k1	73	~ 129
secp256k1 (compressed R)	65	~ 113
secp192k1 (compressed R)	49	~ 92

The computational overhead has been measured by timing 100 executions of the signature and verification methods. This time also includes creating the packets data structure and writing/reading it from a buffer, but does not include transmitting the packet via the radio. The packets are chosen to be 142 bytes long, whether a signature is added or not by using different sized payloads. Results are shown in Table II.

TABLE II
TIME REQUIRED FOR CREATING/READING PHYSICAL 100 PACKETS

elliptic curve	create	read
secp256k1	$\sim 11.5s$	$\sim 34.6s$
secp192k1	$\sim 6.8s$	$\sim 24.9s$
none / unsigned	$\sim 5.5ms$	$\sim 6.6ms$

The use of the *secp192k1* elliptic curve parameters reduces the computational effort for creating and signing a packet by about 40%. Meanwhile, the time spent on extraction and verification of the signature is only reduced by about 30%. The lower reduction for reading and verifying the signature may be due to a larger portion of the time being spent on parsing the buffer containing the raw packet bytes, in contrast to writing the buffer.

To measure the power requirements, a USB power meter is used to monitor the ESP32's power consumption while performing 100 transmissions of messages – spaced 5 seconds apart – with and without signing. Those measurements include everything from creating, (if activated) signing/verifying, transmitting/receiving, to idling in between transmission. The measured values are shown in Table III for the transmitting (tx) and receiving (rx) side and show that increasing complexity of the signature algorithm increases power consumption, as well as that the receiver in general consumes more power.

TABLE III
POWER USED FOR 100 SIGNING AND VERIFICATION RUNS

elliptic curve	tx	rx
secp256k1	$41mWh$	$62mWh$
secp192k1	$36mWh$	$56mWh$
none / unsigned	$33mWh$	$53mWh$

The receiver's higher base power consumption can be explained because the radio has to be powered the whole time in order to listen to and receive messages, while the transmitter can deactivate the radio while it is not transmitting. The difference in power consumption between using *secp256k1* elliptic curve parameters and *secp192k1* is about 10%, which can still be significant in applications, but not as noticeable as the timing comparisons from Table II with 30% to 40% differences lead to believe. A probable explanation for the lesser impact in power consumption may be the fact that CPU cycles are cheaper power-wise than radio transmissions in general and, as a result, the time spent computing only has a lesser influence on overall power consumption.

VII. CONCLUSION

After summarizing the typical challenges of securely and reliably identifying nodes and authenticating their messages in mesh networks, this paper presented an prototype, which applies the key principles of self-sovereign identity management, i. e., decentralized identifiers and verifiable credentials, adapted to the specifics of MANET environments. No permanent Internet connection is required, the limited resources of the devices and the energy-efficiency of operations are considered, and the typical low-bandwidth and simplex / broadcast communication properties are respected.

Given the challenge of keeping data overhead to a minimum, elliptic-curve-based Schnorr signatures were chosen and adapted to LoRa-based communication. The defined packet format and the implementations of signature generation and verification have been evaluated theoretically under security aspects and practically regarding performance, with a focus on the signature's data overhead, the resulting time-on-air impact, and the increase in power consumption. The results demonstrate that verifiable message authenticity comes at a cost, but for example using the elliptic curve *secp192k1* yields an overhead that results in an acceptable trade-off for communication in current real-world mesh networks.

Our next steps include provisions for creating and presenting VCs as well as looking into adapting more parts of the standardized DIDCom protocol to increase compatibility. This compatibility could for example be used to connect the ESP32 LoRa radios to smartphones in order to build a SSI based messaging system in the likes of Meshtastic.

References

[1] R. Baig, R. Roca, F. Freitag, and L. Navarro, "guifi.net, a crowdsourced network infrastructure held in common," *Computer Networks*, vol. 90, pp. 150–165, Oct. 2015.

[2] R. Ramanathan, C. Servaes, W. Ramanathan, A. Dusia, and A. S. Sethi, "Long-Range Short-Burst Mobile Mesh Networking: Architecture and Evaluation," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Jun. 2019, pp. 1–2.

[3] K. Hester, "GitHub – Meshtastic," Meshtastic, Aug. 2020.

[4] International Telecommunication Union, "Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band," no. ITU-R M.1371-5, Feb. 2014.

[5] P. Takemoto and T. Jones, "Fact Sheet – Automatic Dependent Surveillance-Broadcast (ADS-B)," https://www.faa.gov/news/fact_sheets/news_story.cfm?newsid=7131, Jun. 2010.

[6] J. Y. Kim, R. Holz, W. Hu, and S. Jha, "Automated Analysis of Secure Internet of Things Protocols," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, ser. ACSAC 2017. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 238–249.

[7] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici, "A Survey on Secure Communication Protocols for IoT Systems," in *2016 International Workshop on Secure Internet of Things (SIoT)*, Sep. 2016, pp. 47–62.

[8] N. Saxena, G. Tsudik, and J. H. Yi, "Efficient Node Admission and Certificateless Secure Communication in Short-Lived MANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 158–170, Feb. 2009.

[9] K. Hamouid and K. Adi, "Efficient certificateless web-of-trust model for public-key authentication in MANET," *Computer Communications*, vol. 63, pp. 24–39, Jun. 2015.

[10] H. Niavis, N. Papadis, V. Reddy, H. Rao, and L. Tassiulas, "A Blockchain-based Decentralized Data Sharing Infrastructure for Off-grid Networking," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2020, pp. 1–5.

[11] Mazi Project, "Developing a DIY networking toolkit for location-based collective awareness," http://www.mazizone.eu/, 2018.

[12] M. Selimi, A. R. Kabbinale, A. Ali, L. Navarro, and A. Sathiaseelan, "Towards Blockchain-enabled Wireless Mesh Networks," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, ser. CryBlock'18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 13–18.

[13] Guifi.net, "guifi.net - Open, Libre and Neutral Telecommunications Network," https://guifi.net/en, 2020.

[14] M. Balduzzi, A. Pasta, and K. Wilhoit, "A security evaluation of AIS automated identification system," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14. New York, NY, USA: Association for Computing Machinery, Dec. 2014, pp. 436–445.

[15] LoRa Alliance, "LoRaWAN™ 1.1 Specification," *LoRa Alliance*, 2017.

[16] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, Jan. 1991.

[17] H. Morita, J. C. N. Schuldt, T. Matsuda, G. Hanaoka, and T. Iwata, "On the Security of the Schnorr Signature Scheme and DSA Against Related-Key Attacks," in *Information Security and Cryptology - ICISC 2015*, ser. Lecture Notes in Computer Science, S. Kwon and A. Yun, Eds. Cham: Springer International Publishing, 2016, pp. 20–35.

[18] Y. Seurin, "On the Exact Security of Schnorr-Type Signatures in the Random Oracle Model," in *Advances in Cryptology – EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Berlin, Heidelberg: Springer, 2012, pp. 554–571.

[19] D. R. Hankerson, S. A. Vanstone, and A. J. Menezes, *Guide to elliptic curve cryptography*. New York: Springer, 2003.

[20] D. Reed, J. Law, and D. Hardman, "The Technical Foundations of Sovrin," p. 26, Sep. 2016.

[21] A. Tobin, "Sovrin: What Goes on the Ledger?" p. 12, Sep. 2018.

[22] C. Saraf and S. Sabadra, "Blockchain platforms: A compendium," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, May 2018, pp. 1–6.

[23] D. Hardman, "Aries RFC 0005: DID Communication," Nov. 2019.

[24] J. Preshing, "Hash Collision Probabilities," https://preshing.com/20110504/hash-collision-probabilities/, May 2011.

[25] D. Chadwick, D. Longley, M. Sporny, O. Terbu, and D. Zagidulin, "Verifiable Credentials Implementation Guidelines 1.0," Sep. 2019.

[26] Tobias, "TobleMiner/lmic-esp-idf," Aug. 2020.

[27] J. Lovejoy, "metalicjames/cschnorr," Sep. 2020.

[28] P. Aimonen, "nanopb/nanopb," nanopb, Jan. 2021.

[29] H. Dobbertin, A. Bosselaers, and B. Preneel, "RIPEMD-160: A strengthened version of RIPEMD," in *Fast Software Encryption*, G. Goos, J. Hartmanis, J. Leeuwen, and D. Gollmann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, vol. 1039, pp. 71–82.

[30] J. Nakajima and M. Matsui, "Performance Analysis and Parallel Implementation of Dedicated Hash Functions," in *Advances in Cryptology — EUROCRYPT 2002*, ser. Lecture Notes in Computer Science, L. R. Knudsen, Ed. Berlin, Heidelberg: Springer, 2002, pp. 165–180.

[31] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The First Collision for Full SHA-1," in *Advances in Cryptology – CRYPTO 2017*, ser. Lecture Notes in Computer Science, J. Katz and H. Shacham, Eds. Cham: Springer International Publishing, 2017, pp. 570–596.

[32] F. Liu, C. Dobraunig, F. Mendel, T. Isobe, G. Wang, and Z. Cao, "Efficient Collision Attack Frameworks for RIPEMD-160," in *Advances in Cryptology – CRYPTO 2019*, ser. Lecture Notes in Computer Science, A. Boldyreva and D. Micciancio, Eds. Cham: Springer International Publishing, 2019, pp. 117–149.

[33] D. R. L. Brown, "SEC 2: Recommended Elliptic Curve Domain Parameters," *Standards for Efficient Cryptography 2 (SEC 2)*, vol. 2, p. 37, Jan. 2010.

[34] A. Jivsov, "Compact representation of an elliptic curve point," Mar. 2014.