SAND2020-5993C

Secure Energy Constrained LoRa Mesh Network

Derek Heeger[1,2] [0000−0002−4666−8538], Maeve Garigan[3] [0000−0003−4242−3901], Eirini Eleni Tsiropoulou[2] [0000−0003−1322−1876], and Jim Plusquellic[2] [0000−0002−1876−117X]

[1] Sandia National Labs, Albuquerque NM, USA
[2] University of New Mexico, Albuquerque NM, USA
{heegerds,eirini,jplusq}@unm.edu
[3] Roper Solutions Inc, Las Cruces NM, USA
maeve@ropertag.com

Abstract. LoRa (Long Range) is a low-power wide-area network technology well-suited for Internet of Things (IoT) applications. This work is motivated by a cattle monitoring application that involves the communication of Global Positioning System (GPS) and accelerometer sensor data over an ad-hoc LoRa network. Battery-powered sensors are attached to cattle and communicate with each other to form a mesh network capable of relaying data between cattle and a base station. Cattle are highly mobile and travel long distances to unpredictable locations, and the utilization of an ad-hoc LoRa mesh network enables effective monitoring through infrequent data updates communicated over long distances. The proposed work incorporates security into the LoRa mesh network while maintaining an ultra-low energy profile for the battery-powered sensors. Updates are accomplished using GPS-enabled time synchronization and a concurrent transmission property inherent to LoRa. Lightweight authentication and encryption techniques are proposed to prevent spoofing and to provide confidentiality in the message exchanges between cattle and the base station. The energy consumption of the proposed secure implementation is compared with an equivalent insecure implementation. Several mesh network device distributions are constructed and compared to determine the average and worst-case energy consumption of the proposed scheme.

Keywords: LoRa · Mesh Networks · Cattle Monitoring.

1 Introduction

The Internet of Things paradigm has led to a large expansion in consumer applications and services, and a corresponding requirement that such applications run on energy efficient, battery-operated devices. The energy used in the transmission of information between devices and access points represents a significant fraction of the total energy consumption, especially in rural operational environments where transmission over long distances is required. Low-power wide-area network (LPWAN) technologies, such as LoRa, Sigfox, and Narrowband IoT (NB-IoT) [1], offer a distinct advantage in such environments.

LoRa is a closed source protocol managed by the LoRa alliance that uses chirp spread spectrum (CSS) modulation to achieve high noise immunity at the expense of slow data rates. LoRa has configurable parameters such as spreading factor (SF), bandwidth, and error coding rates which enable trade-offs between range and noise immunity [2]. LoRa is typically used as the transport layer for LoRa wide-area networks (LoRaWAN), where the IoT devices can communicate directly to LoRa gateways. Given these salient characteristics, LoRa has recently received a great deal of interest from the industrial, educational, and amateur radio community. LoRa has already been applied in a variety of IoT applications, such as smart cities, industrial IoT, animal tracking and farming, smart metering, and environment monitoring [3].

Motivated by these observations, we propose a secure and ultra-low power protocol that leverages the LoRa mesh network, that is designed to accommodate applications with infrequent data updates. The proposed framework is suitable for battery powered devices that are GPS-enabled or time synchronized using an initialization phase with precision oscillators. Applications that use the proposed topology will benefit from its increased data transmission range and improved reliability in packet delivery while experiencing only incremental energy consumption usage.

1.1 State of the Art & Motivation

A wireless mesh network (WMN) is a concept where client devices alternately act as message relays improving system-wide reliability by enabling the efficient routing of information from a generic source to a generic destination [4]. Mesh network architectures exist for many IoT standards, including WiFi, Bluetooth, and Zigbee [5]. Reactive and proactive routing protocols have been proposed within WMNs to determine how the system discovers message routing. In the proactive routing protocols, the IoT devices maintain routing tables to represent the entire network topology, while in the reactive ones, a multi-hop route is created only upon specific request, thus, the routing overhead is reduced. Typical WMNs have a route discovery phase to generate an internal forwarding table based on the message destination. The routes remain valid until the IoT device's status changes (e.g., change of position or offline mode), causing a maintenance phase. Due to the complexity associated with maintaining and updating the routing tables, WMNs can consume significant amounts of energy.

Focusing on the LoRa specific implementations, a variety of routing-related architectures has been proposed in the recent literature. A LoRa mesh network is introduced in [6] to monitor underground infrastructure. The proposed architecture consists of stationary sensors using GPS time synchronization to minimize their energy consumption and dedicated relay nodes to transmit data to the LoRaWAN gateways. The authors develop a novel LPWAN based on the physical LoRa layer (LoRa PHY), and they show that it overcomes the transmission limitations (i.e., medium-range underground connectivity and time stamping of data packets) of the LoRaWAN standard for underground applications. Towards addressing the problem of single-hop communication supported by LoRa,

a multi-hop routing protocol is introduced in [7] supporting multi-hop networking between LoRa gateways to extend the coverage. The proposed multi-hop routing protocol exploits the Hybrid Wireless Mesh Protocol (HWMP) and the Ad-hoc On-Demand Distance Vector Routing (AODV), and remains compatible with LoRaWAN operation principles. The problem of signals attenuation or blocking by obstacles is addressed in [8], where a mesh network using the LoRa PHY is developed and its packet delivery performance is evaluated. The results reveal that the proposed scheme outperforms in terms of packet delivery ratio compared to star-network topologies, however, it is insecure and the LoRa gateways are constantly wall powered, thus, the mobility aspect is not examined.

The multi-hop routing in LoRa mesh networks has been further examined in [9, 10] by considering the concurrent transmissions of the IoT devices. In [9], the authors consider concurrent transmissions in a LoRa mesh network and they improve the packet delivery rate by introducing timing offsets between the relaying packets. In [10], a scalability analysis is performed for large-scale LoRa networks. The authors' research findings show that LoRa networks, adopting static settings and a single sink, face scalability issues, while the development of multiple sinks and the dynamic communication parameter settings can conclude to scalable solutions. Similarly, in [11], the concept of forwarder-node is introduced and included in between the IoT device and the gateway to improve the range and quality of LoraWAN communications. This work has been extended in [12], where the IoT devices transmit packets to intermediate relay nodes, which retransmit the packets to the LoRa gateways following the Destination-Sequenced Distance Vector (DSDV) routing protocol.

1.2 Contributions & Outline

Despite the significant advances that have been obtained in each of the aforementioned areas in isolation, limited research work has been performed in securing the LoRa mesh networks, while at the same time extending the effective communication range of the IoT devices to the base station. Our paper aims exactly at filling the aforementioned research gap by introducing a custom ad-hoc network scheme based on the LoRa PHY towards enabling the ultra-low power operation of the examined IoT system, while still having advanced capabilities, such as mesh networking and firmware updates. Additionally, we propose a packet structure that supports and enables the data privacy and authenticity, while adding minimal packet overhead.

This work is motivated by a realistic free-range cattle monitoring application, where sensors (i.e., IoT devices) collect location and activity data, and then transmit the collected data to a base station [13]. These sensors reside on the ear tags of the cows providing little space for a battery, thus, creating a severely size and energy constrained scenario. Moreover, the cattle are highly mobile and capable of traveling long distances, making the communication link unpredictable. The cattle act as large dielectrics and absorb a considerable portion of the radio-frequency (RF) transmission energy, thus, concluding to undesirable phenomena, such as a cow that is nearby a base station to appear much

farther away. Therefore, a direct communication link between the cattle and base station can be unreliable in certain scenarios.

In our proposed framework, the LoRa mesh network uses adjacent cattle as relays to transmit the collected data to the base station. It should be noted that typical mesh networks consume significant energy, are unable to tolerate mobility, or are insecure. Our novel concept avoids those issues by using GPS time synchronization to minimize power consumption and taking advantage of the concurrent transmission property inherent to LoRa.

The key scientific contributions of our work, that differentiate it from the rest of the existing literature, are summarized as follows.
1) We introduce a novel LoRa mesh network concept by jointly exploiting the GPS time synchronization and concurrent transmissions to efficiently collect data from low-power sensors that are spread in a large area and are characterized by high mobility levels.
2)The LoRa mesh network concept has a secure implementation involving lightweight encryption and authentication which has been neglected in prior LoRa mesh networks. This prevents malicious packet sniffing, data spoofing, and intelligent denial of service (DoS) attacks.
3) A realistic application has been developed to support the cattle monitoring in an open space area. The proposed framework has been evaluated and modeled to show energy consumption for a variety of device distributions. The protocol is being implemented on custom sensor boards.

The rest of the paper is organized as follows. Section 2 provides a detailed description of the mesh network. Section 3 provides an analysis of the average and worst case performance of the design. Section 4 provides a detailed description of the security process and section 5 concludes the paper.

2 Cow Mesh

We propose our own LoRa mesh network optimized for cattle monitoring applications. The network was designed for a custom sensor platform that has a GPS and accelerometer collecting health and activity data as seen in Fig. 1a and 1b. The sensor has a small capacity battery that recharges via solar panel, and can transmit and receive LoRa data. Minimizing energy consumption while still ensuring reliable packet delivery is critical. This technique is applicable to any energy-constrained GPS-enabled platform that does infrequent data updates.

The cattle sensors typically idle in an ultra-low power consumption mode unable to receive any incoming messages, therefore they must know when to enable their receiver. Receiver current consumption can be greater than 5mA which is an unacceptable power drain for a significant period of time. This limitation adds complexity because all devices must wake up at the same time to be able to operate as a mesh network. Our system uses GPS time synchronization in order to coordinate when to wake up to begin communications. This ensure reliable time keeping even through poor oscillator tolerances and power outages. An event is defined as a complete exchange where the base station attempts to

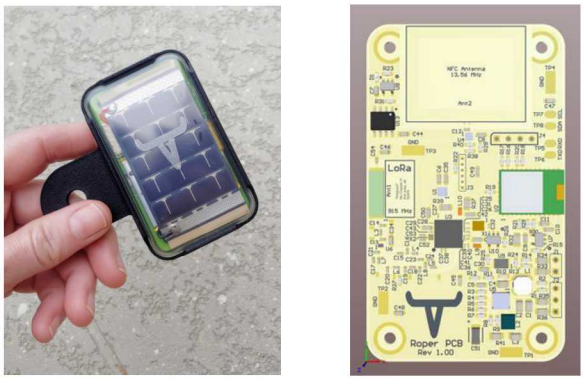

Fig. 1: (a) Roper cattle sensor in housing. (b) Graphic of the sensor board inside of the PCB.

collect data from all cattle sensors. An event occurs repetitively at some predetermined time interval which will likely be every 1-4 hours. The devices will be checking their location more frequently than the event rate which is correspondingly re-synchronizing each sensor's time. Every event can be divided into rounds which can further be subdivided into synch and data frames. These will be defined in detail in the following sections.

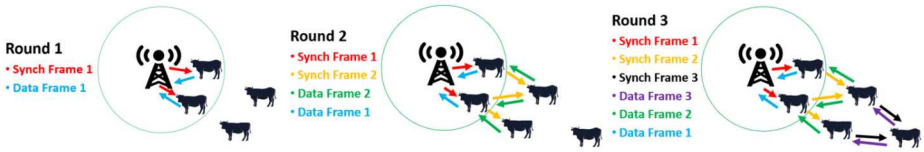

Fig. 2: Mesh Motivation

2.1 Components

An event is divided into rounds which are defined as a single attempt to collect data from sensors with a set number of mesh hops. For example, round 1 involves collecting data from sensors available with direct communications to the base station as depicted in Fig. 2. Rounds 2 and 3 involves collecting data using one or two mesh hops respectively also shown in Fig. 2. Therefore each round progresses one hop further searching for cattle. A round is composed of synch packets which progress away from the base station and data packets which progress towards the base station as depicted in Fig. 3. The primary role of the synch packets are to communicate which cattle have been heard from and to serve as the

route finding mechanism. The data packets contain the location and activity information relevant for each cattle. Each data window is subdivided into C slots which is the total number of cattle in a herd.

The number of rounds required depends on the distribution of devices. For example, if every device was within range of the base station, it would only require 1 round. If there were C cattle spaced equally at the communication range boundaries, it would require C rounds. In practicality, not all devices will be accessible every event so there must be a stop condition to avoid useless rounds. The stop condition can be a complete round where no additional cattle have been heard from or it can be a fixed number of rounds.

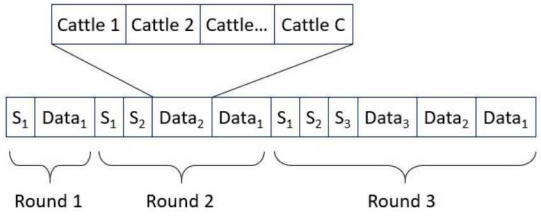

Fig. 3: Packet sequence

2.2 Packet Structure

A detailed breakdown of the synch packet can be seen in Fig. 4. There is a base ID which defines which base station it belongs to (note that only cattle associated with that base ID will relay data). The round number is incremented at the end of each round and the hop is incremented every hop within each round. There is a bit-mapped field that indicates which cattle have been heard from by the base station. The size of this depends on the number of cattle in the herd. The packet is terminated with a CRC for reliability. The secure version is 20-bytes larger than the insecure version because it requires a digital signature and time stamp. The digital signature is the first portion of the packet encrypted using AES-128 and is computed with the hop count always equal to 0. This technique is known as AES-CMAC. The 4-byte time stamp is required to prevent malicious replay attacks. The security theme will be addressed in more detail in section 4.

The data packet for an individual cattle is shown in Fig. 5. The base ID, Hop Count, and CRC serve the same purpose. The herd ID indicates which cattle the data came from. The cattle data contains 25-bytes of location and activity data in the insecure case. The data is 32-bytes (same data plus a 4-byte time stamp and 0-padding to work with AES-128) for the secure case and is encrypted.

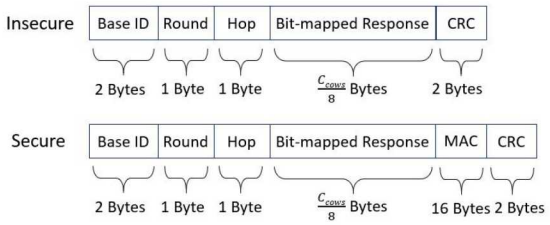

Fig. 4: Synchronization packets

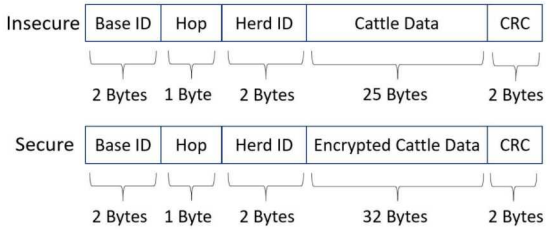

Fig. 5: Data packets

2.3 Transmit and Receive Logic

The base station operation is fairly simple with the following steps:

1. Create a valid synch packet with the bit-mapped response as 0's.
2. Transmit and then wait for data packets to arrive. The wait time increases after each round.
3. Record which cattle it has heard responses from and resend the synch frame with the updated bit-mapped response. The base station will check for message authenticity before updating the bit-mapped response.
4. Repeat steps 2 and 3 until the stop condition.

The sensor transmit and receive logic is more complicated than the base station. The goal is to minimize transmit and receive time to minimize power consumption. After waking up, the receiver listens for a valid synch packet. If it does not hear one in the synch window, it will return to sleep until the next synch packets are being sent. The number of hops it takes to reach the receiver is defined as the device hop count. After receiving the synch packet, it only listens for data packets when the hop count is higher than its own. The device must store other sensors' received data packets in case it must re-transmit them. An example of the receive logic is shown in Fig. 6 for the scenario where a device would hear the second synch frame. The device would listen during the green portions and sleep during the orange.

The sensor needs to transmit both synch and data packets to contribute to the mesh. Upon hearing the synch packet, the device compares the hop count to the round number. If the round number is greater than the hop count, then the device will transmit a synch packet during the next synch window. The

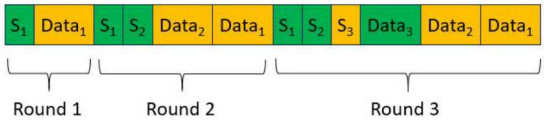

Fig. 6: Receive Logic

device needs to check the bit-mapped response field to see if the base station has received its own data. If not, the device will transmit its data during its time slot. Note that the data window is subdivided into C slots. Only the data from cattle 1 should be transmitted in the cattle 1 slot. Other sensors can transmit cattle 1 data in slot 1, but only as a relay mechanism. Concurrent transmissions occur when multiple devices are relaying the same data or synch packet. When the device has heard data packets from sensors with a hop count greater than itself, it will re-transmit them during the appropriate window. For example, if cattle 1 receives the second synch frame, it will transmit its own data in data frame 2 slot 1. If it then received data from cattle 2 during data frame 3 slot 2, it would relay cattle 2 data during frame 2 slot 2. This is depicted in Fig. 7.

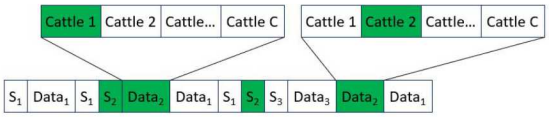

Fig. 7: Transmit Logic

2.4 Time Definitions

For the rest of the analysis, we use the following time definitions depicted in Fig. 8.

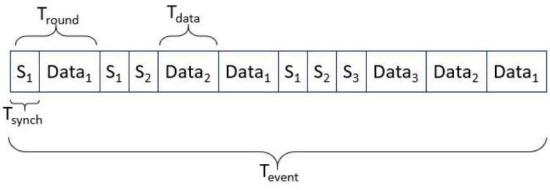

Fig. 8: Diagram of Time Definitions

– T_{synch} is the time for the synchronization packet.
– T_{data} is the time for the entire data window. Note that an individual cattle data packet is T_{data}/C
– T_{round} is the sum the synch and data packets. T_{round} increases each round.
– T_{event} is the total event time which is the sum of all rounds to collect data from all sensors.

3 Analysis

LoRa settings can be changed to vary the transmit time in exchange for communication range. Fig. 9a shows the effect on the spreading factor and bandwidth to send a synch packet for the secure and insecure case. Fig. 9b shows the total time of the data window based on number of cattle over a various LoRa settings for both the secure and insecure case.

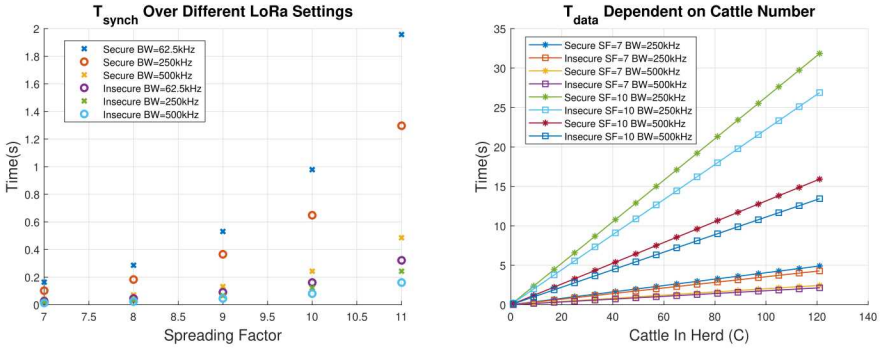

Fig. 9: (a) Time of synch packet over various LoRa Settings, (b) Time for a data round dependent on cattle number.

The energy consumption is defined generically in the following where h is the hop count, R is the total number of rounds in the event, C is the total number of cattle in a herd, and n is the number of devices with the same or less hop count.

$$E_{device}(h,R,n) = (T_{synch}(R-h)+T_{data}(C-n))P_t+(T_{synch}h(\frac{h+1}{2}+R-h)+T_{data}(R-h)/C)P_r$$

The best case energy consumption scenario for a give device is defined as:

$$E_{best} = T_{data}P_T + 2T_{synch}P_r$$

The worst case energy consumption for a give device occurs when sensors are spaced equally in a line requiring all C rounds.

$$E_{worst} = C((T_{data} + T_{synch})P_T + (T_{synch} + (C-1)T_{data})P_r)$$

The total energy consumption for all sensor nodes is defined by simply adding all the individual contributions.

$$E_{total} = \sum_{c=0}^{C} E_c$$

In order to illustrate the performance of this mesh topology, we simulated the performance over a 1-D distribution of cattle. This was done because it illustrates the worst case performance compared to the 2-D scenario in practicality. The analysis provided assumes average Pt of 330mW, average Pr of 15.9mW, C=64, SF=9, BW=250kHz, and error coding rate of 1.25.

We first calculated the energy performance of a linear distribution of cattle as indicated in Fig. 10a. The total energy consumed per device located at each hop in the mesh can be seen in along with the breakdown consumption due to transmit and receive power. The devices located closer to the base station consume more power because they must receive more packets to transmit and relay on more data. The peak power consumption for the devices at the first hop was 3.5J which is reasonable. For reference, a small rechargeable lithium-ion battery has 200mAh of capacity which correlates to 2.7 kJ. This means that our sensor could do this mesh operation over 700 times without recharging. A comparable scenario is shown in Fig. 10b where the sensors are distributed equally. The linear distribution is more practical for cattle monitoring applications because the base stations will be located near water sources.

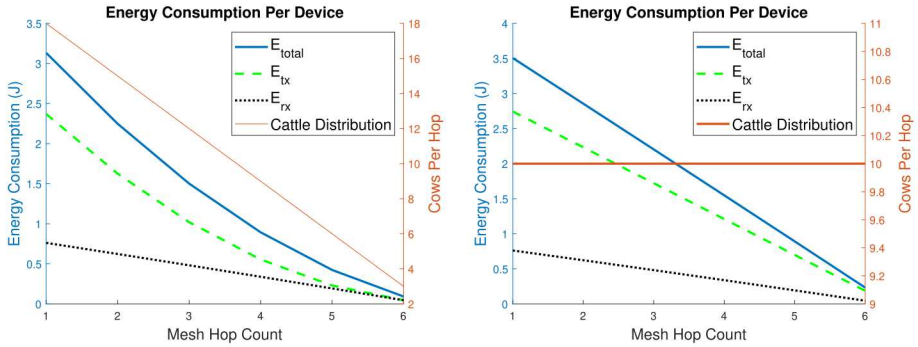

Fig. 10: (a) Energy consumption if cattle have linear distribution (b) Energy consumption if cattle have an equal distribution.

It is also important to assess how energy consumption scales with increased cattle numbers and spreading. Fig. 11a shows how the energy consumption compares as the maximum hop count required increases, indicating the sensors are more spread out. It assumes there are a fixed 64 cattle in a herd and then compares the energy consumption as they distribute across more hops. The device

average and maximum consumption is compared for both linear and equal distributions. This plot indicates a linear dependence on energy consumption with maximum hop count, so the maximum hop count could be a tool for limiting energy consumption (at the risk of not hearing from devices). The total energy consumption for all devices dependent on cattle number distributed across 6 hops is shown in Fig. 11b which scales linearly.

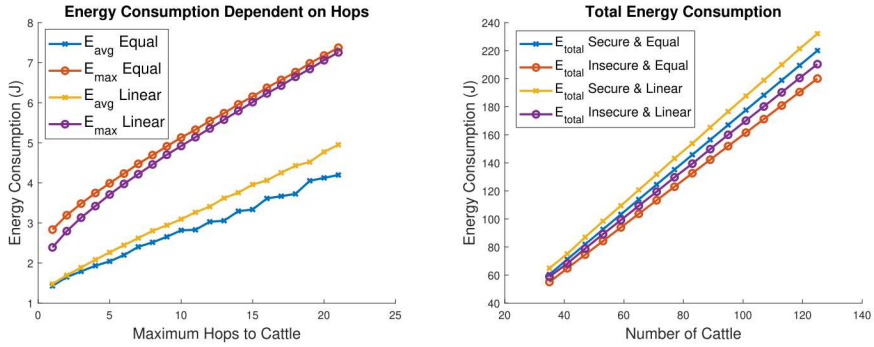

Fig. 11: (a) Energy consumption per device dependent on number of hops (b) Energy consumption for same number of hops with different number of cattle.

4 Security

The primary goals of a secure system are to ensure data privacy and to prevent false impersonation of devices. Data privacy is ensured by using encryption and false impersonation is prevented using a message authentication code (MAC). Spoofing could be used to report incorrect data into the base station or carry out an intelligent denial of service (DoS) attack.

There are two keys programmed into every device which include a herd key K_h and cattle specific key K_c. K_c is unique to every device but K_h is common to the entire herd. Each key is 128-bits because all security operations use AES-128. The base station is able to securely access a server with all the keys.

Privacy is ensured by using data encryption with K_c. Cattle within the same herd do not know each others encryption keys, so they are unsure of the specific data that they are forwarding in the mesh network. The 7 additional bytes used in the encrypted data packet include a time stamp which assists the base station in detecting false data.

The synch packet is authenticated with a MAC using K_h. Therefore, every device can validate the authenticity of the fields in the packet. If this were not the case, a simple DoS attack would be to rebroadcast a synch packet that says the base station has heard from everyone triggering all devices to go to sleep until

the next event. It is critical that there is a unique element that contributes to the MAC which is why the 4-byte time stamp is required. If this were not the case, the system would be vulnerable to replay attacks from prior captured packets since the same MAC would still apply. The data packet is also authenticated using a MAC with K_h.

A malicious actor will not be able to decrypt any cattle specific data without compromising individual security keys. If a single device is physically compromised, an adversary could use invasive techniques or side-channel analysis to try to identify the device specific keys [14, 15]. In the event they compromised K_h, they would be able to impersonate data coming from the base station to the sensors. They would be unable to send false data back to the base station because they do not know K_c for all the other cattle. This prevents them from encrypting the data such that it decrypts with a valid time stamp causing the base station to disregard the packet as garbled data.

In the event that a device goes entirely off line (i.e. cattle or cattle tag went missing), it would be prudent to update the herd key. This would prevent the adversary from having sufficient time to extract the existing keys. Secure re-keying is the process of selectively updating K_h for every device. This is only needed if a new device is brought into a herd or if K_h is suspected to be compromised. Distributing $K_{h,new}$ would be accomplished by sending a packet encrypted by the base station with K_c to each device. This update requires a different set of packet exchanges not defined in this work.

5 Conclusion

In this work, we proposed a secure mesh network concept suitable ultra-low power consumption. This architecture is optimized for devices with GPS time synchronization that do infrequent data updates. The protocol is described in detail including packet structures and transmit/receive logic. The system is modeled to assess performance over a variety of cattle distributions and densities, and proves suitable for energy constrained devices. The security is assessed and provides adequate privacy and authenticity with minimal energy overhead. The cattle sensors have been developed and future work includes the performance evaluation on a ranch to confirm the reliability of the design.

Acknowledgments

The research of Dr. Eirini Eleni Tsiropoulou was conducted as part of the NSF CRII-1849739.

We would like to thank Kevin Nichols for the hardware design support and Mike Partridge for the motivation.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC,

a wholly-owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525

References

1. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: A comparative study of lpwan technologies for large-scale iot deployment. ICT express **5**(1) (2019) 1–7
2. SX1276, L.: 77/78/79 datasheet, rev. 4 (2015)
3. Sarker, V., Queralta, J.P., Gia, T., Tenhunen, H., Westerlund, T.: A survey on lora for iot: Integrating edge computing. In: 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), IEEE (2019) 295–300
4. Hossain, E., Leung, K.K.: Wireless mesh networks: architectures and protocols. Springer (2007)
5. Cilfone, A., Davoli, L., Belli, L., Ferrari, G.: Wireless mesh networking: An iot-oriented perspective survey on relevant technologies. Future Internet **11**(4) (2019) 99
6. Ebi, C., Schaltegger, F., Rüst, A., Blumensaat, F.: Synchronous lora mesh network to monitor processes in underground infrastructure. IEEE Access **7** (2019) 57663–57677
7. Lundell, D., Hedberg, A., Nyberg, C., Fitzgerald, E.: A routing protocol for lora mesh networks. In: 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), IEEE (2018) 14–19
8. Lee, H.C., Ke, K.H.: Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation. IEEE Transactions on Instrumentation and Measurement **PP** (03 2018) 1–11
9. Liao, C.H., Zhu, G., Kuwabara, D., Suzuki, M., Morikawa, H.: Multi-hop lora networks enabled by concurrent transmission. IEEE Access **5** (2017) 21430–21446
10. Bor, M.C., Roedig, U., Voigt, T., Alonso, J.M.: Do lora low-power wide-area networks scale? In: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. (2016) 59–67
11. Velde, B.: Multi-hop lorawan: including a forwarding node (2017)
12. Dias, J., Grilo, A.: Lorawan multi-hop uplink extension. Procedia computer science **130** (2018) 424–431
13. : Revolutionizing beef production
14. Skorobogatov, S.: Flash memory 'bumping'attacks. In: International Workshop on Cryptographic Hardware and Embedded Systems, Springer (2010) 158–172
15. Zhou, Y., Feng, D.: Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. IACR Cryptology ePrint Archive **2005**(388) (2005)