BcWAN: A Federated Low-Power WAN for the Internet of Things (Industry track)

Mehdi Bezahaf
Worldline
Lille, France
mehdi.bezahaf@worldline.com

Gaëtan Cathelain
Worldline
Lille, France
gaetan.cathelain@worldline.com

Tony Ducrocq
Worldline
Lille, France
tony.ducrocq@worldline.com

ABSTRACT

This paper introduces BcWAN, a roaming solution for an IoT LoRa-based network that allows IoT end-devices to deliver data to their home network going through foreign [1] gateways.

Our architecture removes the central core network and replaces it with a blockchain that handles the network access control. Any gateway in the system can communicate directly with another gateway in a peer-to-peer manner while maintaining confidentiality, integrity and soundness. Our work solves the fair exchange problem introduced in such architecture where no third party is involved thanks to a combination of encryption and specific blockchain techniques like custom script operators. We implement a proof of concept of the BcWAN architecture to gather an insight of the performance of the solution. We outline that BcWAN itself does not add any significant overhead to a near real-time IoT application by presenting preliminary test results.

ACM Reference Format:
Mehdi Bezahaf, Gaëtan Cathelain, and Tony Ducrocq. 2018. BcWAN: A Federated Low-Power WAN for the Internet of Things (Industry track) . In *19th International Middleware Conference Industry (Middleware '18 Industry), December 10–14, 2018, Rennes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3284028.3284036

1 INTRODUCTION

After Internet and mobile communications, the Internet of Things (IoT) is designated to be the third wave of information technology. ABI Research [1], Cisco [2] and Ericsson [3, 4] have all presented constructive reports about IoT and the findings are unequivocal with a forecast of rapid growth and a comparable increase diversity of IoT applications and services. This burst of the IoT technologies has opened up the potential to build an ecosystem of billions of connected devices [5] supporting a new large set of application domains including but not limited to smart city, such as smart metering, smart parking, vehicle fleet tracking, and smart street lighting to name but a few.

[1]Foreign gateway means that a user/company X's device can use a foreign gateway that is a property of Y to deliver its own data, where $X \neq Y$.

Middleware '18 Industry, December 10–14, 2018, Rennes, France

ACM ISBN 978-1-4503-6016-6/18/12…$15.00
https://doi.org/10.1145/3284028.3284036

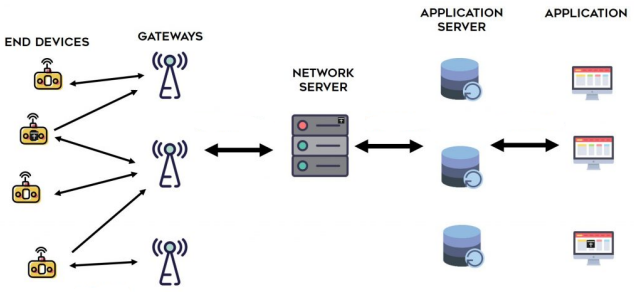

Figure 1: LoRaWAN End to End Network Overview

Low-Power Wide Area Networking (LPWAN) technology offers *long-range communication* with *low energy consumption*, which empowers new kinds of services and applications. Due to these features, this technology fits perfectly the needs of the Internet of Things (IoT) and Machine-to-Machine (M2M) networks [6]. An IoT device usually is characterized by a small electronic equipment that consumes little energy to send limited amount of data per time, which grants them the ability to operate for many years on a simple battery. Nonetheless, LPWAN technologies share a common drawback which is the overall *limited capacity of transmission*. The duty cycle inferred by some of the used unlicensed radio spectrum combined with a low bandwidth hampers the transmission capabilities, restricting LPWANs use cases. LPWAN technologies combine low data rate and robust modulation to achieve communication over several kilometers. Such range allows for limited deployment requirements by lowering the number of required antenna/gateway [6–9].

Different LPWAN implementations exist on the market (Sigfox [10], WaveIoT [11], or NB-IoT [12] to name but a few), and LoRaWAN [13] is one of the most successful LPWAN technologies [14]. It can be due to its open source software support and the ability for anyone to build its own private network. In fact, LoRaWAN explores the unlicensed radio spectrum as Wi-Fi does, which means that anyone can use the radio frequencies without having to pay fees for transmission rights. This flexible deployment strategy has attracted industries and companies to invest on this technology. It allows them to either use an existing public network or setup their own private if they are concerned by the ownership of the network. LoRaWAN also comes with an ease of installation, completely free of intellectual property and royalty costs (which is not the case of Sigfox), and with the most resilient privacy/security level.

With the low power aspect (multi-year life, coin cell operation) and the long range feature, LoRaWAN has drawn attention of industrial actors to answer diverse needs. Whether in term of asset management like asset tracking and monitoring (airports, car lots, construction sites, warehouses, retail), energy and land use optimization, pallet tracking, shipping containers; smart cities like smart meters, parking sensors, street lights control, waste management; or smart buildings and home like water leak detection, termite traps, and smart home sensors.

The architecture of LoRaWAN (Fig. 1) consists in a set of end-devices that sense and collect data and transmit it to a gateway within their radio range; a set of gateways that act as an access point and send packets to a network server through a backhaul connection; a core network (Network Server) that processes all received packets from gateways and directs them to an application server; and finally, a set of application servers that handle the customer application and treat the data.

Deploying a private network is convenient in term of privacy/security aspect but even if a LoRa gateway can cover a large Km-area, it requires considerable resources to independently cover your full devices' operating area (e.g. countries). As described above, if a company want to deploy its own private network, it needs to setup gateways, application servers and most importantly the core network (Fig. 1). So an interesting question can be: *Can we use foreign gateways to forward our data while maintaining privacy and security standards?*

In this paper, we present BcWAN, a federated and decentralized platform that allows different parties to access and collect their nodes' data through foreign gateways while maintaining confidentiality and security standards. We take away the single actor deployment scheme and replace it with a blockchain-based architecture that handles the network control access, and which allows gateways to communicate directly through a Peer-to-peer network. We consider that a *node* belonging to a *recipient* can send some collected data to the *recipient* passing through a foreign *gateway* while :

- the *gateway* and the *recipient* entities may not trust each other
- insuring $confidentiality$ of the data
- *gateways* should not be able to receive more data than what it participates in the network
- and the *gateway* wants to be rewarded for the data transferred

By doing this, we create an interesting outcome that leads us to tackle two interesting challenges in this paper:

(1) Knowing the communication limits of LoRa, how to ensure that the message received by the *gateway* from the *node* intended to the *recipient* is not going to be disclosed.
(2) Once the *gateway* has received the encrypted message from the *node*, how to ensure a fair exchange between the *recipient* and the *gateway*.

The remainder of this paper is organized as follows. Section 2 outlines the background of this paper article and introduces the reader to the concepts of a blockchain. In Section 3, we present the current literature and explain the architectural decisions. Section 4 introduces our protocol and design. We describe the performance of our solution in Section 5 and discuss the results and design choices in Section 6.

2 BACKGROUND

In parallel with the striking growth of the IoT, the Blockchain technology drew more and more attention. As LPWAN services are growing towards centralized single-proprietary infrastructures, we can see the potential of mutualizing hardware to allow broader coverage. Sharing of such infrastructure can be enforced through binding contracts between parties. That would involve renting or supervising access and would be enforced by law. But with the emergence of Blockchain, and by issuing contracts through, e.g., Bitcoin scripts, companies now have a way to ensure that all players are following the rules in a common network without the need of a third party and/or lawyers.

The Blockchain is a tamper-proof decentralized ledger, where the rules are enforced through a consensus between the users. Blockchain technology has first been introduced by Satoshi Nakamoto [15] as a mean to offer a solely electronic version of currency without a third party (e.g. central banks).

His proposal, Bitcoin, is considered as the first Blockchain-based cryptocurrency to be implemented. From this work, several cryptocurrency solutions emerged such as Ethereum [16] or ZCash [17]. As Blockchain also empowers certification services via Proof of Ownership, Proof of Existence or Proof of Integrity, this work led to non-financial oriented proposals. Namecoin [18] uses Proof of Ownership to provide a decentralized Domain Name System (DNS). It prevents censorship and surveillance from governments as the association table is stored in a decentralized manner while assuring Proof of Ownership of a domain.

Our work, BcWAN, heavily relies on Blockchain scripts. Blockchain scripts are a list of instructions in a transaction which describes how the next user can gain access to the output of a transaction (i.e. the cryptocurrency available in the output of a transaction). Scripting is done through a non-Turing-complete stack-based language. This mechanism especially empowers consumers to do direct payment to one another by using Elliptic Curve Digital Signature Algorithm (ECDSA) signatures and keys. Any user owning funds can emit transactions whose output contain a certain value (i.e. amount) directed to another Bitcoin address. Each output is locked and can only be redeemed by the user who owns the private key behind the Bitcoin address the output is destined to. Through these techniques, the Blockchain becomes a tamper-proof decentralized ledger, where the rules are enforced through a consensus between the users. Beside direct payment, Bitcoin scripts are malleable enough to build more complex rules for transactions. One could for example lock the output of a transaction to the preimage of a sha256 hash. This way, the user that desires to unlock the amount would have to reveal the preimage of the given sha256 hash. Blockchain also gained interest in the IoT field thanks to the aforementioned feature.

3 RELATED WORK

Wörner and Bomhard proposed a general data exchange mechanism [19] through the Bitcoin network. They use Bitcoin scripting capabilities to enable 1) traceability, 2) payment of data through

Bitcoin and 3) storage of data inside a Blockchain. This method enables efficient, reliable and verifiable storage of data via Bitcoin. Because of the limitations in the consensus of Bitcoin, data storage is limited to 40 bytes. Their work focuses on selling data while our work focuses on selling networking capabilities.

Our solution was inspired by Zero Knowledge Contingent Payment (ZKCP), which was first introduced in 2011 by Gregory Maxwell [20]. As we aimed to build a mutualized network, the fair exchange was a core need and ZKCP was a solution to our problem. Although introduced in 2011, the method was only theoretical as no Zero-Knowledge proof mechanism fitted the need of ZKCP at the time. Later, zero-knowledge Succint Non-interactive ARgument of Knowledge (zk-SNARKs) rendered the implementation of ZKCP and made them practically available. In 2016, Gregory Maxwell realized the first ZKCP on the main Bitcoin network [21]. In the context of IoT, computational power is limited on the edge network. As Gateways in LoRaWAN are positioned at this level, we wanted to limit the impact of the Blockchain on those devices by limiting storage and computational needs. At the time of writing, current zk-SNARK implementations are relatively (i.e. comparably with an embedded architecture) both memory and computational heavy [22], which is why we moved away from this approach.

The Things Network [23] is a LoRaWAN-based global, crowd-source, open and free Internet of things data network that allows everyone to deploy their own gateways to be used by their own end-devices and by other users. PicoWAN [24] is a LoRaWAN-based collaborative, low power wide area network for connected objects. The network infrastructure is built from ground up by the user community installing PicoWAN's gateways inside buildings that will establish a wireless link between connected objects and the Internet. Both solutions, cited above, answer perfectly our open question but both approaches have a unique single actor that deploys and operates the network and the gateways. ARCHOS [25] is the only player that administers the PicoWAN network and sells the PicoWAN's gateways (PicoPlugs). For the Things Network's approach, it is a little bit flexible, meaning that you can use your own gateway with their semi-open-source code. However, the core network still under their control, like operating, monitoring, and optimizing the network; managing, configuring and troubleshooting the gateways; granting access to users, requested network quality of service, and billing. In their work Durand et al. [26] propose a solution where a Blockchain acts as an activation server in order to create a full P2P network without Network Server. An interesting aspect of their work is that it does not require modification of the end devices to work, neither the LoRaWAN gateways. However, their solution does not incentive *gateways* of the network and thus it reduces users interest in deploying *gateways* . Their work is complementary to ours as it is design for altruistic users in mind like The Things Network.

4 ARCHITECTURE

4.1 Requirements

The goal of BcWAN is to create a shared LoRaWAN infrastructure allowing different parties to deploy and use the network without relying on a single network operator and without having to deploy the whole network on their own. The idea of BcWAN is that it does not rely on a trusted third party, which may be difficult to define in such situation. Indeed, BcWAN does not require any centralized server nor any trusted third party[2]. In order to prevent abuses on the network, BcWAN relies on crypto-currency and micro-transactions. Each time an actor of the network sends a message through one of his nodes, he must pay for the delivery of the message. When an actor dispatches a message thanks to one of his gateway, he receives a few crypto-currency as a reward for processing the message.

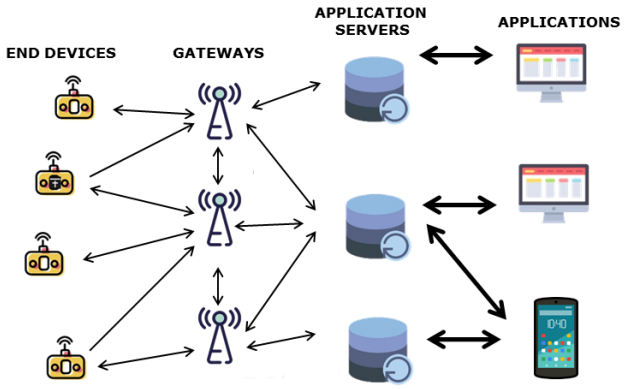

Figure 2: BcWAN Network Overview

The problem raised with this principle is that we face a fair exchange problem between the actor that wants to retrieve the data from the dispatcher and the dispatcher that wants to receive the crypto-currency in exchange for the data delivery. The following sub-sections describe the BcWAN architecture and how we have solved this problem.

4.2 Overview

The overall BcWAN architecture (Fig. 2) shows that there is no more network server, comparing to the basic LoRaWAN architecture (Fig. 1), and that *gateways* communicate directly with each others and with the application servers. For the sake of simplicity, we assume that each actor of the network possesses only one gateway. With several gateways per actor, each actor will have to elect one of his gateways as the *master* gateway [3].

In BcWAN, when a gateway receives a message, instead of sending it to the *network server*, it looks up in the blockchain the network address (IP address) to whom the message is intended. The gateway then sends the message to this recipient which will in its turn send it to the right *application server*. The choice of the *application server* is not different to what we have in legacy LoRaWAN network.

In BcWAN there are two main challenges that will be addressed in the next sub-sections:

(1) Address directly a *gateway* from another *gateway*,
(2) Solve the fair exchange problem.

[2] except a blockchain which may be considered as a trusted third party.
[3] The *master* gateway is the gateway to whom all the actor's devices have to address their data to.

4.3 Addressing gateways directly

When a *gateway* receives a message, it needs to send it to the right recipient thought TCP/IP network (the Internet). To do this the *gateway* needs to know the IP address of the recipient or its Fully Qualified Domain Name (FQDN). In BcWAN, we stick to IP addresses because the system we put in place is similar to Domain Name Server (DNS) and would be quite redundant if using FQDN.

When a gateway has to deliver a message to another gateway, it can only rely on information sent by the node (i.e. the sensor). The node may not directly know the IP address of the recipient, mainly because the latter can change if the recipient gateway is moved on another network. However the node knows a unique identifier which is actually the blockchain address of the recipient (*@R*). Each recipient that is ready to receive messages on a given IP address must create a blockchain transaction containing the information relative to its IP address. The gateway which needs to deliver the message will then do a lookup in the blockchain to find the IP address associated to this blockchain address. A communication can then be established using a TCP/IP socket between the sender and the recipient to negotiate the exchange of the data.

4.4 The fair exchange

The fair exchange problem in BcWAN comes from the following dilemma in case of malicious parties:

(1) The gateway could receive the payment but never deliver the data
(2) The recipient could receive the data but never send back the payment

We can see that if the gateway sends the data first, then the recipient can take advantage of the situation and not pay for the service. If the recipient sends the payment first, the gateway could not deliver the data and be payed without providing the service. This is what is called the fair exchange problem. This kind of problem is especially seen in the context of digital exchange where parties does not know each other and are anonymous.

A solution for this problem could be the usage of reputation. If the recipient pays for the data first it should receive the data in exchange. The gateway does not have many advantages to not deliver the data but in case it does not, the recipient can alter the reputation of the gateway. Also if the gateway delivers the data but receives a bad notation from the recipient, it can also alter its reputation. This solution reduces the probability of misbehavior but does not eliminates the problem. In BcWAN, we propose a solution where both parties are guaranteed to get what they are owed.

In BcWAN, we need to guarantee the following properties:

(1) The confidentiality of the data
(2) The integrity of the data
(3) Authenticity of the data
(4) Proper payment to the gateway if the recipient receives the data
(5) Retrieving the data by the recipient in case of payment

In order to guarantee the confidentiality of the data, the node and the recipient share a symmetric key (*K*). The message can be encrypted with this key on the node and deciphered on recipient side. This step is mandatory if confidentiality of the data is required.

For the integrity and mainly for the authenticity of the data the node also signs the encrypted message (*Em*) and the ephemeral public key (*ePk*) with a secret key (*Sk*). The node and the recipient must also share a secret key (*Sk*), on the node, and a public key (*Pk*), on the recipient. A provisioning phase is therefore needed in order to load the necessary keys on the node.

To guarantee that the payment is done if and only if the data is sent by the gateway to the recipient we use the capabilities provided by the Blockchain scripts which are also known as smart contracts. The idea is the following:

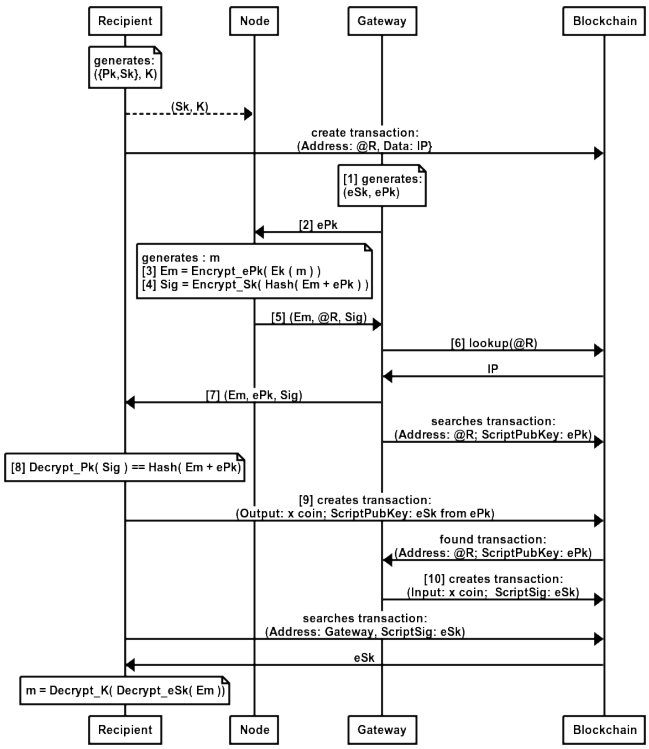

Figure 3: BcWAN Message exchange sequence diagram.

(1) After receiving a first request from the *node* (not illustrated in Fig. 3), the *gateway* generates an ephemeral public (*ePk*) and secret key (*eSk*) pair. This pair will only be used for this message.
(2) The *gateway* sends the ephemeral public key (*ePk*) to the *node* using LoRa.
(3) The *node* double encrypts the message (*m*) using fist the symmetric key (*K*) and then the ephemeral public key (*ePk*) and obtains (*Em*).
(4) The *node* signs the result of the encryption (*Em*) and the ephemeral public key (*ePk*) using his own secret key (*Ska*). The signature is called *Sig*.
(5) The *node* sends the message encryption (*Em*), the signature (*Sig*) and the *recipient* address (*@R*) to the *gateway* using LoRa technology.
(6) The *gateway* retrieves the IP address of the *recipient* in the Blockchain as explained in Section 4.3.

(7) The *gateway* sends the message encryption (*Em*), the ephemeral public key (*ePk*) and the signature (*Sig*) to the *recipient* using TCP/IP.
(8) The *recipient* verifies the authenticity of the message thanks to the signature (*Sig*).
(9) The *recipient* creates a transaction in the Blockchain with a given output (fixed or negotiated with the gateway) and a specific script. The script requires to provide the secret key *eSk* associated with the public key *ePk* in order to unlock the amount provided in the transaction.
(10) When the *gateway* receives this transaction, it creates a new transaction using the output of the previous one and provides the ephemeral secret key *eSk*. The output of this transaction is not important but should be intended to the *gateway* itself.

This exchange protocol (Fig. 3) requires a specific type of transaction in order to be resilient to attacks. Given that the funds contained in step 9 are locked until the correct secret key is given, it is necessary for the transaction to also be time-locked. For Bitcoin, the *OP_CHECKLOCKTIMEVERIFY* script operator was introduced to deal with time-locked problems. This operator enables funds from a transaction to be locked for a given amount of blocks. The *recipient* can then specify that the funds are locked until a) the associated private key is revealed or b) a given amount of time has passed. We also introduced the *OP_CHECKRSA512PAIR* script operator in order to verify that the Private key given by the *gateway* is the one that matches the public key in the transaction. The *OP_CHECKRSA512PAIR* script operator has been implemented using the *VerifyPubKey* method of *RSA_PrivKey* class from OpenSSL. The listing 1 gives the complete script used in BcWAN.

Listing 1: Ephemeral Private Key Release Script

```
1  <rsaPubKey>
2  OP_CHECKRSA512PAIR
3  OP_IF
4  OP_DUP
5  OP_HASH160
6  <pubKeyHash>
7  OP_EQUALVERIFY
8  OP_ELSE
9  <block_height+100>
10 OP_CHECKLOCKTIMEVERIFY
11 OP_VERIFY
12 OP_DUP
13 OP_HASH160
14 <buyerPubkeyHash>
15 OP_EQUALVERIFY
16 OP_ENDIF
17 OP_CHECKSIG
```

5 PROOF OF CONCEPT

In order to test the feasibility of BcWAN, we implemented a proof of concept using a couple of software layers. The first layer manages the LoRa exchanges and is deployed on a Nucleo-144 (STM32F746) running as the *node* and a Raspberry Pi equipped with a LoRa shield (RFM95 LoRa module) running as the *gateway* . Once the *gateway* receives the encrypted message from the *node* through LoRa exchanges, it switches to the second layer, which deals with the Blockchain interactions. We define those layers as the two main modules, the *LoRa module* and the *Blockchain module*.

5.1 Modules overview

For the *LoRa module*, we modified the work of C. Pham [27] to meet our needs on our *gateways* . We used Multichain for the *Blockchain module* [28]. Multichain is a fork of Bitcoin v10.0 which provides interesting features from a Blockchain testbed point of view such as modifying the average mining time, the size of a block or the consensus in a Blockchain. Those parameters impact the theoretical maximum number of transactions per second that a Blockchain can process, thus the overall performance of it.

Multichain does not compile on ARM processors, and due to the fact that our *gateway* is running on a Raspberry PI, we had to split the *gateway* 's modules into two hardware units: a Raspberry PI for the *LoRa module* and a Linux virtual machine for the *Blockchain module*.

In order to insure the confidentiality of data, our *node* uses the Advanced Encryption Standard (AES) [29, 30] with Cipher Block Chaining (CBC) mode to encrypt the data. The original message (plaintext) is split into a fixed block size (16 bytes). If the plaintext length is less than 16 bytes, we add some padding. The first block is XORed with a random vector of the same size (IV). The obtained result is encrypted with our AES-256 symmetric key shared between the *node* and the *recipient*. The number of ciphertext blocks is equal to the number of plaintext blocks. We assume that our message's length will be less than 16 bytes (temperature, humidity level,...). Hence, our obtained ciphertext is about 16 bytes. Additionally to the ciphertext, the *node* has to send the random IV to the recipient in order to be able to decrypt the message. We end up having 34 bytes (Fig. 4) to send to the recipient, which can be easily encrypted using the asymmetric public key of the *gateway* .

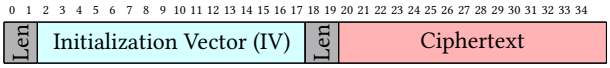

Figure 4: The encrypted message using AES-256 and its IV.

As described in Section 4.4, in one hand, the *node* has to sign the result of the encryption (*Em*) and the ephemeral public key (*ePk*) generated by the *gateway* . We use the RSA-512 encryption for this matter. We discuss this choice in Section 6. Using the shared asymmetric key with the *recipient* (*Sk*), we insure to the *recipient* the authenticity of the message and that (*ePk*) was the genuine ephemeral public key used in the process. On the other hand, the *node* has to double encrypt the message (*m*) using the symmetric key (*K*) and then the ephemeral public key of the *gateway* (*ePk*). Given the chosen encryption methods, we effectively have a predefined minimum payload of 128 bytes, 64 bytes for the double data encryption and 64 bytes for the signature.

Multichain operates as a daemon responding to JSON-RPC requests. We encapsulated Multichain around our BcWAN daemon implemented in Golang. Our daemon listens on a given port for requests from foreign *gateways* . It then interacts with Multichain API to 1) create the transactions, 2) sign the transactions and 3)

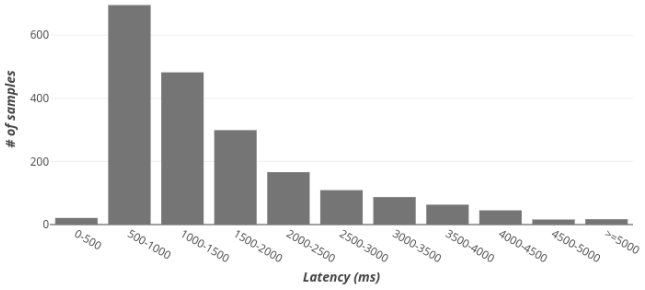

Figure 5: BcWAN process latency (*without block verification*)

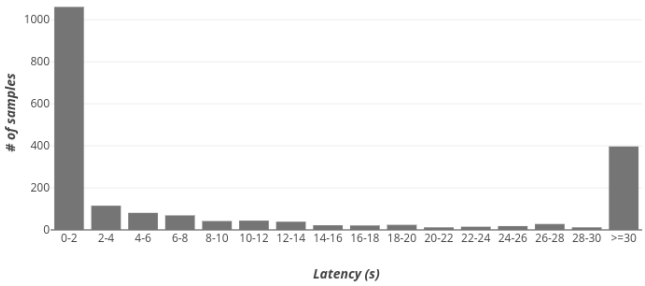

Figure 6: BcWAN process latency

send the transactions. The daemon is also responsible for broadcasting the *node* IP and retrieving other *gateways* IPs. We used the *OP_RETURN* script operator to do so, which allows to publish arbitrary data inside the output of a transaction. On start-up, each *node* retrieves the recent blocks from other *nodes* and scans their content for foreign *gateways* IPs. We also modified Multichain's code in order to introduce a new operator which checks that a private RSA-512 key matches a public RSA-512 key. This allows us to infer a payment for the disclosure of a private RSA-512 key.

5.2 Performance overview

We chose to draw an overview of the impact of the *Blockchain module* in BcWAN. To do so, we used PlanetLab [31], which is a distributed platform for deploying and evaluating planetary-scale network services. We chose 5 PlanetLab *nodes* with similar specifications. Each *node* has 4 CPU Cores and 512 MB of RAM. Given 128 bytes of payload and 4 bytes of length header, we simulated 30 sensors per *node* at a 1% duty cycle using a LoRa Spreading Factor level 7, effectively giving us a theoretical maximum of 183 messages per sensor per hour. An AWS EC2 instance is used as a master *node* only to 1) bootstrap the *nodes* and 2) mine blocks. Mining is disabled on the PlanetLab *nodes* .

We measured the overall latency of 2000 exchanges in BcWAN using the aforementioned setup. Multichain advertises a transaction throughput of up to 1000 tx/s (transaction per second) in its latest version. We saw different results during our experiments as the block verification made the Multichain daemon stall and become unresponsive for extended periods upon each block arrival. We chose to differentiate the exchange latency of BcWAN to Multichain's performance by disabling block verification. We see that, if there is no block verification (Fig. 5), the mean latency of a full exchange is *1.604s*, from the first message from the *gateway* to the decryption of the message by the *recipient* . On the other hand, with block verification (Fig. 6), the mean latency of a full exchange is of *30.241s*.

6 DISCUSSION

We show in the previous section that BcWAN limits the overhead added to LoRa communications to a few seconds. This does not take into consideration the block verification process of Multichain which is not the subject of this paper. We aspire that other implementations or future Blockchain technologies may solve this issue. The mean latency allows for close to real-time communications in a Peer-to-peer manner while maintaining security standards. The added processing power required at the edge nodes can be mitigated by offloading computational heavy tasks to application servers (e.g. mining, block verification). The presented results do not take into account the edge geolocation nature of Peer-to-Peer communication. In a real world environment, a sensor has higher chances to communicate with a Gateway that is geolocated closer to his origin deployment. The network latency can thus be decreased between co-located foreign Gateways and lower the data retrieval latency.

In BcWAN we chose to allow the foreign *gateway* to not wait for confirmation of the *recipient* transaction before providing the ephemeral private key. This can be a security threat as a malicious user could double spend this transaction. Bitcoin advises users to wait a minimum of 6 confirmations, and because Bitcoin has an average block time of 10 minutes, effectively wait 60 minutes after each transaction. If the *recipient* double spends the first transaction, the *recipient* can retrieve the ephemeral private key necessary to decipher the encrypted data without rewarding the foreign *gateway* . The addition of a confirmation time on the exchange protocol to prevent double-spending implies an added latency. Even though we use a two-transaction protocol, confirmation on the second transaction is non-necessary as the *gateway* has no incentive to try to double spend the second transaction. Our choice for the Proof of Concept is motivated by the fact that we wanted to separate the performance of BcWAN from the performance of the underlying blockchain technology.

We chose RSA-512 as method to encrypt our data due to the size limit of the payload that can be sent on the LoRa network, which is highly constrained. This lowers the security as RSA-512 can be brute-forced [32][33] but the amount to spend in order to decrypt the data is (nowadays) much more than the value that the foreign *gateway* is asking to reveal the ephemeral private key. For application where this may be a problem it is possible to use higher levels of encryption but messages will be lengthier on the LoRa network.

The Proof-of-Work is not suitable for edge nodes to run the blockchain as this is a computational power based method of election. Other methods such as Proof-of-stake [34] do not rely on computational power and thus can help to further close the gap of the blockchain to the edge nodes. While a Blockchain network based on Multichain is able to process thousands of transactions per second, it is ultimately unsuited for processing large quantities

of data. The scalability issues are out of the scope of this paper as BcWAN could be implemented on another blockchain technology.

7 CONCLUSION

We introduce a new architechture for LoRa-based networks allowing the deliverance of data through foreign parties without the need of a trusted third party. This is especially powerful as it allows parties with a shared goal to securely deploy a common network in a fair manner while maintaining security standards. Rules are enforced through a blockchain in order for parties to fairly exchange inside the federated network. Parties that don't participate to the network aren't able to take advantage of foreign property. Our experimental evaluation showed that we maintain the desired security standards while mitigating the induced overhead compared to trustful IoT networks. Overall, we introduce a new way for parties to safely exchange data in a potentially malicious environment without the need of a trusted third party.

REFERENCES

[1] A. Research, Cellular m2m connectivity services - the market opportunity for mobile operators, mvnos and other connectivity service providers, Tech. rep., ABI Research (Feb. 2012).
[2] Cisco-VNI, Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021 white paper, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.
[3] Ericsson, More than 50 billion connected devices, White Paper.
[4] Ericsson, Device connectivity unlocks value, White Paper.
[5] S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara, A. G. Skarmeta, A global perspective of smart cities: A survey, in: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 2013, pp. 439–444.
[6] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, M. Pettissalo, On the coverage of lpwans: range evaluation and channel attenuation model for lora technology, in: International Conference on Intelligent Transportation Systems Telecommunications, Copenhagen, Denmark, 2015, pp. 55–59.
[7] P. Neumann, J. Montavont, T. Nol, Indoor deployment of low-power wide area networks (lpwan): A lorawan case study, in: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, New York, NY, USA, 2016, pp. 1–8.
[8] K. Nolan, W. Guibene, M. Kelly, An evaluation of low-power wide area network technologies for the internet of things, in: International Wireless Communications and Mobile Computing Conference, Paphos, Cyprus, 2016, pp. 439–444.
[9] O. Iova, A. Murphy, G. Picco, L. Ghiro, D. Molteni, F. Ossi, F. Cagnacci, Lora from the city to the mountains: Exploration of hardware and environmental factors, in: International Conference on Embedded Wireless Systems and Networks, Uppsala, Sweden, 2017, pp. 317–322.
[10] Sigfox, Sigfox - the global communications service provider for the internet of things (iot), http://www.sigfox.com.
[11] WAVIoT, Waviot: Low power wide area network for iot and m2m, http://waviot.com.
[12] Y.-P. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, H. Razaghi, A primer on 3gpp narrowband internet of things, IEEE Communications Magazine 55 (3) (2017) 117–123.
[13] N. Sornin, A. Yegin, Lorawan 1.1 specification, LoRa Alliance, Technical Report.
[14] A. Research, Low-power wide area network market data, Technical Report.
[15] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf (2009).
[16] G. Wood, Ethereum: A secure decentralised generalised transaction ledger (2014).
[17] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash: Decentralized anonymous payments from bitcoin, in: IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2014, pp. 459–474.
[18] Namecoin wiki, https://wiki.namecoin.org.
[19] D. Worner, T. von Bomhard, When your sensor earns money: Exchanging data for cash with bitcoin, in: ACM International Joint Conference on Pervasive and Ubiquitous Computing, New York, NY, USA, 2014, pp. 295–298.
[20] G. Maxwell, Zero knowledge contingent payment, https://en.bitcoin.it/wiki/Zero_Knowledge_Contingent_Payment (Nov. 2011).
[21] G. Maxwell, The first successful zero-knowledge contingent payment, https://bitcoincore.org/en/2016/02/26/zero-knowledge-contingent-payments-announcement/ (Feb. 2016).
[22] N. Bitansky, R. Canetti, A. Chiesa, E. Tromer, From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again, in: Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 2012, pp. 326–349.
[23] T. T. Network, The things network - building a global internet of things network together., https://www.thethingsnetwork.org.
[24] PicoWAN, Picowan - lora collaborative iot network., http://picowan.com/en.
[25] ARCHOS, Archos - pioneer in android tablets, portable audio and video player., https://www.archos.com/us-en.
[26] A. Durand, P. Gremaud, J. Pasquier, Resilient, crowd-sourced lpwan infrastructure using blockchain, in: Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Munich, Germany, 2018, pp. 25–29.
[27] C. Pham, Building low-cost gateways and devices for open lora iot test-beds, in: International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Hangzhou, China, 2016, pp. 70–80.
[28] G. Greenspan, Multichain private blockchain, White Paper.
[29] T. Robertazzi, Advanced Encryption Standard (AES), Springer New York, New York, NY, 2012, Ch. 10, pp. 73–77.
[30] R. Pereira, R. Adams, The ESP CBC-Mode Cipher Algorithms, RFC 2451, IETF (Nov. 1998).
[31] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman, Planetlab: An overlay testbed for broad-coverage services, ACM SIGCOMM Computer Communication Review 33 (3) (2003) 3–12.
[32] L. Valenta, S. Cohney, A. Liao, J. Fried, S. Bodduluri, N. Heninger, Factoring as a service, in: International Conference on Financial Cryptography and Data Security, Rockley, Barbados, 2016, pp. 321–338.
[33] D. Goodin, Breaking 512-bit rsa with amazon ec2 is a cinch. so why all the weak keys?, https://arstechnica.com/information-technology/2015/10/breaking-512-bit-rsa-with-amazon-ec2-is-a-cinch-so-why-all-the-weak-keys/ (Oct. 2015).
[34] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: Annual International Cryptology Conference, Santa Barbara, CA, USA, 2017, pp. 357–388.