Secure, Decentralized Energy Resource Management using the Ethereum Blockchain

Casimer DeCusatis, Kulvinder, Lotay
Marist College
Poughkeepsie, NY USA
casimer.decusatis@marist.edu

***Abstract*—While blockchain services hold great promise to improve many different industries, there are significant cybersecurity concerns which must be addressed. In this paper, we investigate security considerations for an Ethereum blockchain hosting a distributed energy management application. We have simulated a microgrid with ten buildings in the northeast U.S., and results of the transaction distribution and electricity utilization are presented. We also present the effects on energy distribution when one or two smart meters have their identities corrupted. We then propose a new approach to digital identity management that would require smart meters to authenticate with the blockchain ledger and mitigate identity-spoofing attacks. Applications of this approach to defense against port scans and DDoS, attacks are also discussed.**

Keywords—Blockchain, cybersecurity, identity, authentication, energy, Ethereum

I. INTRODUCTION

The use of blockchain technologies to provide immutable, distributed transaction ledgers has received significant attention recently as a technology with the potential to disrupt and transform virtually every aspect of global business. Recent examples include the use of distributed ledgers by financial firms [1], tracking import/export requirements for the shipping industry [2], tracking manufacturing raw materials and components in the airline industry [3], tracking royalties in digital entertainment systems [4], and guaranteeing the safety of food chains from farm to table [2]. This widespread interest in blockchain distributed ledgers is accompanied by growing concerns regarding cybersecurity. Both the number and severity of cybersecurity incidents have grown dramatically in the past few years. Many of these attacks against blockchains are related to a lack of rigorous authentication and identity management, including protection against both insider and external threats. Poor identity management has been cited as a factor in recent large financial attacks [5] and record-setting multi-terabit/second attacks against service providers [6]. These and other examples motivate the urgent need for improved device authentication, and the problem will only grow worse as blockchains begin to receive input from large numbers of distributed sensors running applications for the Internet of Things (IoT).

One of the areas being disrupted by blockchain is the electrical utility industry [7]. Traditional electrical grids are top-down systems in which utilities (often with legal monopolies) generate and sell power to consumers. This is changing with the proliferation of low cost solar panels and residential energy storage systems. In 2016 the global market for rooftop photovoltaic panels was nearly $30 Billion, and is expected to have double-digit growth over the next few years [7]. At the same time, the capacity of residential energy storage systems is growing from around 95 MW in 2016 to over 3.7 GW by 2025 [7]. This enables a new form of distributed renewable energy generation and distribution that does not require a centralized, large-scale utility company. Instead, energy is distributed using microgrids, a smaller local network that manages its own power generation and peer-to-peer energy trading. Microgrids can significantly reduce or eliminate the lossy transfer of electricity over large distances, while providing increased flexibility and modularity in comparison to the traditional centralized grid. It has been estimated that there are around 160 active microgrids in the U.S. generating about 1.7 GW of capacity; this is forecast to increase to over 4.3 GW by 2020 [7].

Operation and maintenance of a distributed energy grid relies on the automation of transaction processes, as well as effective management of irregular energy production and storage. This may in part be mitigated by the use of a distributed ledger such as blockchain. Several prototypes are being investigated, including the Transactive Grid project in Brooklyn, New York [8] and a project by TenneT Holdings in Germany [9]. However, the true value of microgrids may come from bringing electricity to areas that are under-served or have lost power due to natural disasters. Following Hurricane Maria in October 2017, most of Puerto Rico was without electricity for an extended time; microgrids are being used to help facilitate disaster recovery [10]. Worldwide, nearly 1 billion people still lack basic access to electricity, with a large number in developing economies lacking centralized grid infrastructure. Decentralized electrical grids are being investigated as a way to enable more widespread electric power availability and promote economic development.

In this paper, we present results from simulating a distributed energy grid on a small campus network, and

978-1-5386-4649-6/18/$31.00

managing the energy transactions using an Ethereum smart contract with a blockchain distributed ledger. We have created original code for a microgrid smart contract, which is used for accounting, transfer and interaction of energy generation and usage between smart electrical meters on different buildings. Our code includes a grid synchronization function, which checks for available energy reserves, monitors available smart meters, and initiates energy transaction requests as required. We have built a front-end dashboard application and tested security mechanisms for energy management smart contracts. We have also developed a DApp (on Ethereum, a DApp is a web application that will interact with smart contracts deployed on the blockchain) to offer decentralized energy management and energy accounting, and a platform for peer-to-peer energy trading. Having established a baseline for smart microgrids, we then investigate the effect of injecting erroneous data into the microgrid from corrupted power monitoring devices. We show that if the identity of power monitoring devices is not secured, it is possible to significantly disrupt overall energy management for the entire microgrid. We then investigate key issues related to identity management on Ethereum, and propose an alternative approach based on digital identity gateways that authenticate smart meters and blockchain ledgers. This approach is also useful in mitigating port scans and port-based attacks against the Ethereum blockchain.

The rest of this paper is organized as follows. After an introduction to the problem in section I, we describe the design of our original Ethereum smart contract code in section II. The blockchain architecture and properties are briefly reviewed in section III. In section IV we present results from simulation of a campus-scale network. Section V discusses cybersecurity considerations, proposed a new digital identity architecture, and presents results from our digital identity system. Finally, section VI summarizes our conclusions and objectives for future work.

II. Ethereum Smart Contract Design

Although it has some mathematical precedents, modern distributed ledger technology underlying blockchains was developed around 2008 [11]. Timestamped transaction records are collected to form a data structure called a block. Each block contains a hash of the prior block and a nonce, forming a digital fingerprint linking the blocks to form a blockchain. We may implement blockchains as a linked list with a hashed pointer. In a public blockchain, the transaction record or ledger is shared among nodes in a distributed peer-to-peer network, and forms an immutable transaction record. Each node performs computations required to add new transactions or blocks, a process called mining or proof of work. Blocks may also contain metadata including timestamps and transaction roots from a Merkel tree [12], and optional data encryption for additional privacy. Transaction chains may represent self-enforcing or self-executing service agreements, called smart contracts. A consensus process enables all nodes to agree on the ledger content, and helps identify when a transaction is invalid or a copy of the blockchain is invalid. Being a distributed database, blockchain is subject to Brewer's Theorem (or the CAP theorem), which states that it is impossible to simultaneously achieve consistency, availability, and partition tolerance [12]. Blockchain provides eventual consistency, i.e. the transaction ledgers from all participants will be consistent after some time interval (usually between milliseconds and seconds). Our current work uses the Ethereum blockchain to manage a smart contract in which tokens represent energy transactions. Ethereum was selected because it has been successfully used in similar energy management projects [8, 9] and because of the relative maturity of its Solidity smart contract language [12]. Further, a public blockchain platform was chosen to facilitate the further development of public "energy credit tokens" [8], such as those based on current Ethereum token standards. Such tokens may be tied to different geographic regions and help facilitate value transfer for peer-to-peer energy trading. We also note that use of a public blockchain circumvents the need for local server infrastructure, enabling this solution to be used in regions where local infrastructure has been damaged by natural disasters, or regions that simply lack pre-existing infrastructure.

Our work proposes several enhancements to Ethereum blockchain security. We introduce a new method for identity-based network security, which extends end-to-end from the client to the blockchain fabric. This is realized by authenticating the first packet of a network connection request using cryptographic identity tokens, which are inserted into the packet header by a software or hardware endpoint and later authenticated by a gateway software at the far end of an untrusted network. This approach, called first packet authentication (FPA), allows us to authenticate packets at the earliest possible time (i.e. when a network connection requires is initially made between two devices). All unauthorized traffic (including port scans) is dropped at the transport level, so the traffic source does not receive any acknowledgment or feedback that might be used for reconnaissance as the first step in a cyberattack. This approach, called transport access control (TAC), helps isolate and protect blockchain services from unauthorized access.

Further, we propose that security policies be implemented at the first packet authentication gateway. In our implementation, based on the Blackridge Technology authentication gateways, we assign each device to one of eight dynamically adjustable trust levels. This can be used to control access between a smart meter and a contract on a shared ledger. Trust levels for different authorized users can be changed based on business requirements or in response to potential insider threats. Audit trails for all authorized and unauthorized connection attempts to the blockchain are maintained and can be easily audited using software to parse the log contents.

III. Simulation Results

Although it is not practical to construct a solar energy distribution system at scale for experimental purposes, we have simulated the use of Ethereum smart contracts for a campus scale application. We have based this simulation on a

This work was supported by the National Science Foundation under CC*DNI Integration (Area 4): Application Aware Software-Defined Networks for Secure Cloud Services (SecureCloud). Award #1541384.

subset of buildings at the Marist College campus, located in upstate New York, allowing for factors such as different energy demands and daylight hours in summer and winter months. We have modeled a system based on ten buildings located on the Marist College campus in upstate New York (these buildings do not have solar panels installed, but were selected for simulation purposes as being representative of a typical campus environment). The campus environment is shown in Figure 1, and a list of the relevant modeling parameters for the ten buildings chosen is given in Table 1. Note that larger buildings have proportionally more space on the roof for solar panels, and therefore more capacity to generate solar energy, but also have larger energy consumption requirements.

Table 1 – List of 10 buildings modeled in the microgrid, including size of solar panel arrays and annual energy consumption requirements

Building Name	**Potential solar panel size (square meters)**	**Energy consumption/year (kilowatt-hours)**
Gartland 1	22,500	425,250
Gartland 2	22,500	425,250
Fontaine	22,500	425,250
Foy 1	13,500	255,150
Foy 2	13,500	255,150
Foy 3	13,500	255,150
Dyson	31,500	595,350
Hancock	33,000	623,700
Lowell Thomas	28,000	529,200
Library	38,000	718,200
TOTAL	**238,500**	**4,507,650**

Figure 1 – Microgrid of ten buildings used to simulate Ethereum based energy distribution network

The software stack architecture used in our simulation is shown in Figure 2. Ethereum implements blockchain using the Solidity smart contract language [12]. The blockchain is hosted on an Ethereum Virtual Machine (EVM). It is accessible via a DApp, which is a service enabling direct interaction between users. Two smart meters can communicate with each other (for example, to negotiate availability of energy storage and cost) using the stack shown in Figure 2. Ethereum DApps will typically interface to smart meters via an HTML/JavaScript web application using a JavaScript API to communicate with the blockchain. The Web3 Javascript API was the main JavaScript software development kit (SDK) that we used to create code that interacts with an Ethereum node. We also used the low level JSON RPC 2.0 interface to communicate with a blockchain node (this API is used by the Web3 JavaScript API).

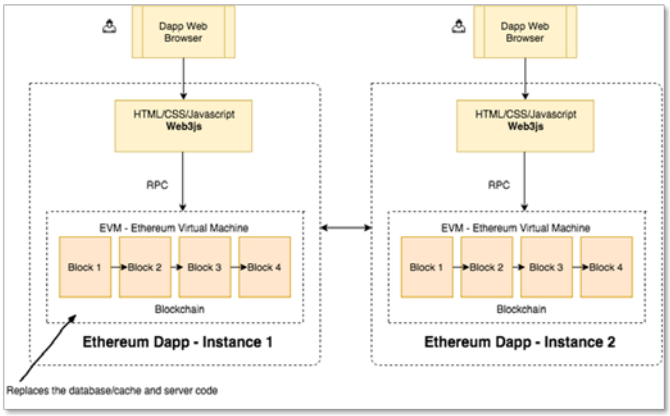

Figure 2 – Software stack architecture for web client (i.e. smart meter) peer-to-peer communication.

Results of the ten building microgrid simulation are shown in Figures 3-7 under various conditions. Each building is modeled as having solar panels proportional to the available space on their roofs, and our smart contract allows energy to be traded between any user on the microgrid. The electricity demand for different buildings varies depending on the building size, time of day, and season of the year. The electricity generation capacity for different building varies depending on the number of solar panels it can support (larger buildings have more roof space), time of day, season of the year, and factors such as the cloud cover (we currently assume cloud cover prohibits solar energy generation 30% of the time during daylight hours; the model assigns this randomly during each week that we simulate). For this simulation, solar panel efficiency was set at 25%, which is consistent with current commercially available photovoltaic systems. During winter months, buildings in the northeast part of the U.S. experience fewer daylight hours than during summer months, and thus have less capacity for electricity generation. For illustrative purposes, we present typical results for a spring day, midway between the winter and summer extremes. We can characterize the microgrid in terms of its energy balance [8]. When the overall microgrid balance is positive, the grid is self-supporting. In other words, the microgrid generates

enough solar energy to meet the demands of all buildings, once energy has been transferred from building generating surplus energy to buildings that are not generating sufficient energy. When the balance is negative, there is more demand for electricity than the microgrid can supply, and the microgrid needs to draw on a conventional external utility grid to make up the difference.

Figure 3 illustrates the energy balance for the microgrid (watts-hour) vs time of day (based on a 24 hour clock), showing the energy generated for each building in the microgrid. During spring in the northeast U.S., there are approximately 14-15 hours of daylight (sunrise at around 5:30 AM and sunset at around 8:30 PM) during which the solar panels can generate energy. During the nighttime hours, solar energy cannot be generated. For convenience, we illustrate the energy generation cycle beginning with sunrise, in two-hour intervals. Figure 3 shows the buildings that generate the most electrical energy at each point in the daily cycle. While more energy is generated from some buildings than from others, the overall energy balance remains positive during daylight hours, then begins to trend negative after sunset as expected. The microgrid requires attachment to an external utility after sunset to meet the requirements of all buildings.

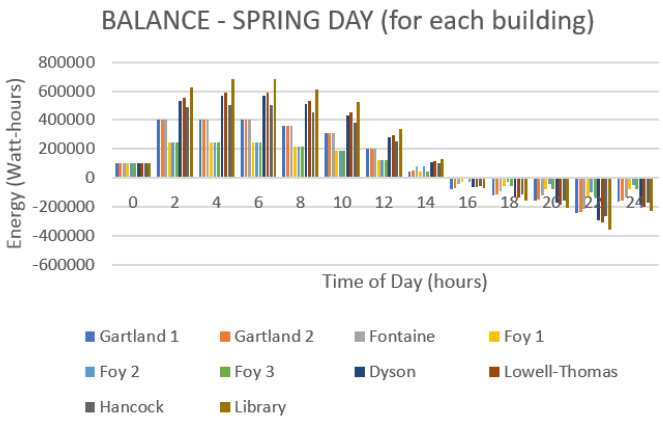

Figure 3 – Energy generated for each building on the microgrid vs time of day

By taking the average energy generation for all ten buildings, we can visualize the total microgrid energy balance at different times of day, as shown in Figure 4. Note that this generally follows the envelope of the individual building bar graphs from Figure 3, as expected. When the smart meters detect that a given building can no longer generate enough energy to meet its own needs, the smart meters can make a request to purchase or transfer surplus electricity from other buildings on the microgrid.

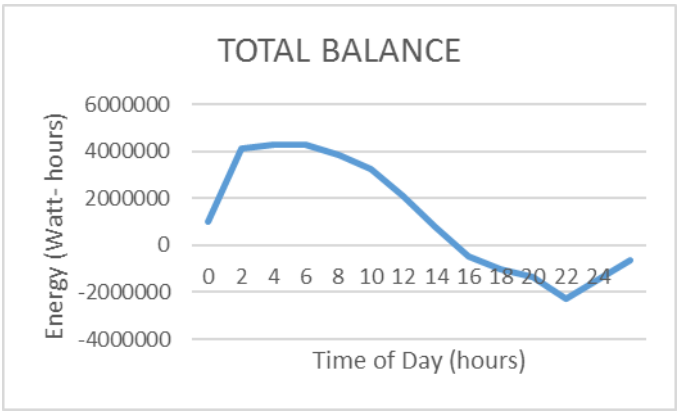

Figure 4 – Total microgrid energy balance vs time of day for a typical spring day

A plot of the energy transfers between buildings is shown in Figure 5. Energy transfer requests commence near sunset, and most of the transfers occur from the larger buildings (which have more solar panels and generate excess capacity) to the smaller buildings (which cannot support enough solar panels to be entirely self-sustaining). As noted previously, in this example the microgrid requires attachment to an external utility after sunset, since even when all available electricity is redistributed the microgrid is not self-sufficient after sunset. This is typical behavior for most commercial microgrids [8, 9] and we will not attempt to further optimize energy transmission, since we are more interested in the effects of cybersecurity breaches on the microgrid infrastructure.

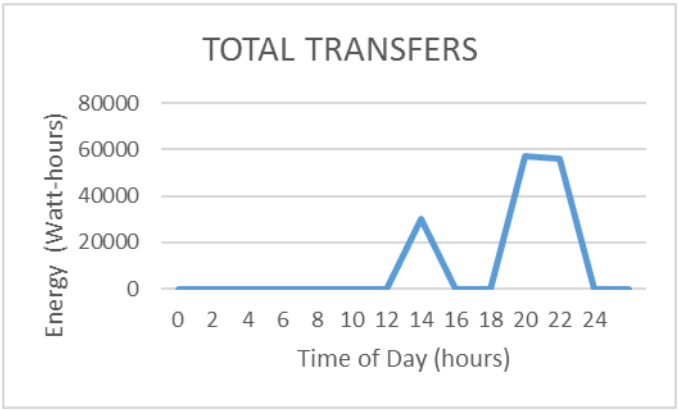

Figure 5 – Simulation results of energy transfers for a spring day vs time of day

The previous figures have illustrated normal operation of a microgrid with blockchain-based energy transfers using smart meters. We will now consider the case where some of the smart meters have had their identity stolen, and the rogue smart meters are spoofing energy requests to the microgrid. We illustrate the case where two buildings (Gartland 1 and 2) are conscripted by malware, and begin to send erroneous energy requests to the grid. Typical results showing the total energy balance with smart meter spoofing are shown in Figure

6. By comparison with Figure 4, we can see that even when a very small number of smart meters are corrupted, the total energy balance is forced into very abnormal behaviour.

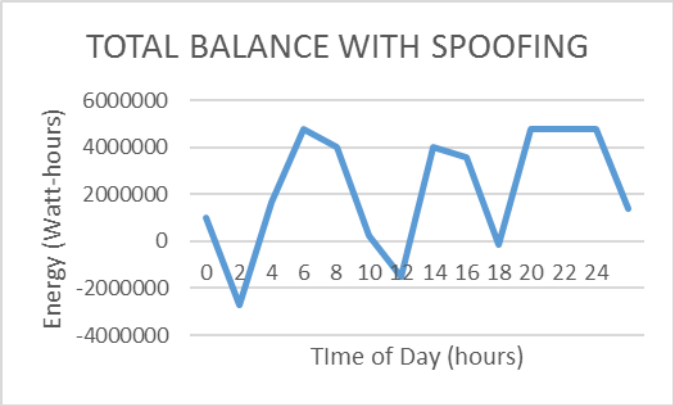

Figure 6 – Energy balance vs time of day when two of the smart meters provide erroneous data (for comparison with Figure 4)

The corresponding energy transfers when smart meters are being spoofed is shown in Figure 7. By comparison with Figure 5, we see once again that it is possible to force significantly abnormal behaviour when only a few smart meters have been corrupted.

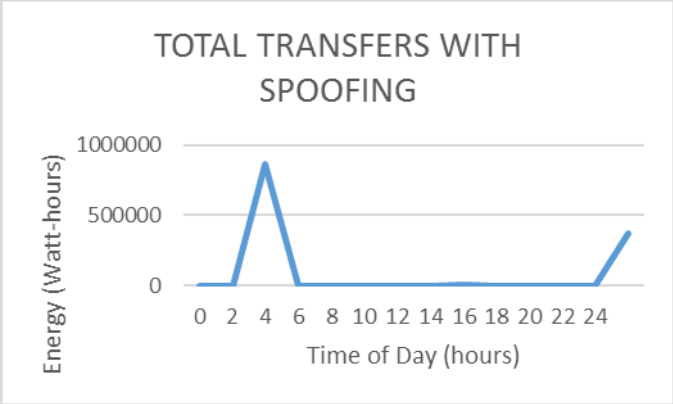

Figure 7 – Simulation results of energy transfers with smart meter spoofing (for comparison with Figure 5).

IV. Security Model

The distributed energy framework faces many of the same fundamental cybersecurity issues as many other IoT applications. In particular, the blockchain used in these experiments lacks a rigorous authentication, nonrepudiation, and identity management architecture. Thus, we are able to spoof the identity of a few smart meters in the grid, and submit erroneous data that significantly disrupts the entire energy balance of the microgrid. This issue can be mitigated if we introduce a new architecture that enforces strict identity management for smart meters. In general, an identity management system requires some form of unique identifier for a device or user, a way for users to ask for and authenticate this identifier, a way to make claims about an identity (such as its name, address, and whether it's allowed to access certain resources), and a way to validate these claims.

Considering a threat matrix for our distributed energy framework, there is a need to better authenticate smart meters to the ledger, so we can protect against fraudulent data sources corrupting the ledger and artificially manipulating energy costs. Similarly, a smart meter digital identity system must perform two-way authentication (i.e. authenticating the ledger to the smart meter) to prevent unauthorized energy distribution requests from the ledger. It is necessary to protect the Ethereum smart contract from both external and internal threat actors. External bad actors may attempt to break into the system as unauthorized users to corrupt data or compromise a smart meter. Internal bad actors are already members of the Ethereum smart contract, who attempt to access unauthorized resources within the smart contract. Both external and internal threat actors may attempt to attack servers hosting the ledger, for example attacking ephemeral open ports using the so-called Bleichenbach-style attacks [13].

The need for a common identity management framework has been articulated by the Decentralized Identity Foundation (DIF), a consortium founded by Microsoft and other companies [14]. Simultaneously, the W3C consortium has been investigating standards for different types of claims about an identity [15]. There have been various approaches to identity management in Ethereum. Recently, the ERC 725 standard [16] has proposed that the address of an Ethereum smart contract could serve as an identifier. This proposal also requires an integrated public key management system, a proxy contract interface for interacting on the Ethereum blockchain, a claims registry (allowing any key pair based account to make public claims on the blockchain about an identity), a way to look up claims in the registry, and a taxonomy of claim types. In addition to its complexity, early drafts of this standard face several issues. The proposed standard defines an interface for identity that can be implemented in many different ways, much like the preceding ERC 20 Token standard [17]; this leads to various potential security issues. Although only smart contracts serve as identities, only key-pair accounts are able to make claims. Each smart meter would have to deploy a contract to the blockchain in order to establish its identity. The burden of identity signature verification falls to the consumer of a claim; if the consumer is a smart contract, it cannot be updated to include new signature types in the future. Scalability and efficiency are also significant concerns. Most ERC 725 operations require two transactions, one for proposing the operation and another for completing it. Adding a claim is a transaction, approving the claim is a second transaction. Thus, the current proposal is inefficient at transaction processing.

We propose an alternative digital identity management system to address these concerns, with significantly less complexity and improved efficiency. Software on the smart meter inserts an identity token (which serves as the identifier)

into packet headers; these packets are later authenticated by a software gateway on the ledger server. The ledger server gateway includes a method to validate smart meter identity claims, such as mapping the identity to an access policy and a trust level. This trust level can be dynamically raised or lowered depending on whether the smart meter is attempting activities which are not allowed by its access control polity (which helps guard against insider threats). The system is bi-directional so that the ledger can also insert identity tokens which are authenticated by the smart meter.

We implement this approach using first packet authentication (FPA) technology with software gateways from BlackRidge Technologies. This is similar to an approach we have described previously for other applications [18-21], so we will only provide a brief overview of the technology. Our proposed identity management system is shown in Figure 8. The smart meter needs to communicate with an Ethereum fabric smart contract over an untrusted network, such as the Internet. One end of our authentication gateway is implemented in software on the smart meter (for example, a BlackRidge Technologies 4.1 client). To facilitate authentication policy management (i.e. claim validation), a node.js server which hosts the other end of the identity authentication gateway is placed in series with the Ethereum blockchain server. When the smart meter attempts a connection to the Ethereum smart contract, a time sensitive identity token (valid for four seconds) is inserted into the first packet of the connection request. After traversing the untrusted network, the token is authenticated by the node.js server. Unauthorized access attempts are discarded, and BlackRidge TAC prevents any further feedback that an attacker might use for reconnaissance. The identity management system on the node.js gateway can immediately blacklist the IP address of an unauthorized bad actor attempting to impersonate a smart meter. As previously noted, the process works in both directions, so that the ledger can also be authenticated to the smart meter. Our original, open source code is available from GitHub [19].

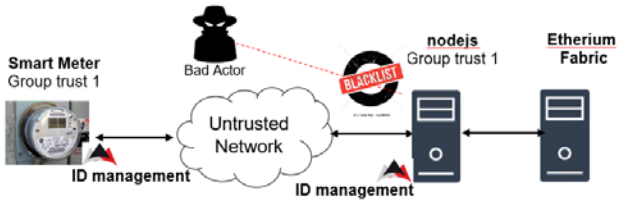

Figure 8 – Proposed identity management approach for smart meters on an untrusted network

The recommended architecture implements Node.js on a separate server from the chain code and its related functions. However, we can implement both the Node.js and chain code within a cloud environment, allowing clients to access this system from anywhere in the world using only a web browser. We note that this implementation also enables automating regulatory compliance auditing.

We have previously completed performance measurements on the token insertion process, and compared them with industry standard benchmarking techniques such as IPerf [20]. We have demonstrated a sustainable data rate of 10.0 Gbit/second and negligible performance impact form token insertion and authentication [21]. We have demonstrated that a Zenmap scan of the node.js application does not pick up open ports when the authentication solution is enabled. Thus, an external threat actor performing reconnaissance cannot identify open ports when our authentication solution is enabled. Further, port scans with authentication enabled do not return any responses at or below the transport layer, since the scanning application is not authenticated. A record of the scan attempt is logged by the authentication gateway for future reference. Note that by inhibiting port scanning and blocking unauthorized access attempts before they reach the Node.js server, denial of service attacks can be mitigated. We have repeated this test multiple times with different clients, to confirm that there is no possibility of unauthorized access because of a compromised authentication key; only users with the appropriate key on a specific computer can access the blockchain service.

In our next penetration test, we demonstrated that the authentication system effectively stops Bleichenbach-style attacks [13] against the Ethereum blockchain. These attacks utilize discrepancies in Transport Layer Security (TLS) error messages to obtain the pre-master secret key private RSA key used by TLS to decrypt sensitive data. While RFC 5246 Section 7.4.7.1 [22] provides recommendation on defending against this attack, TLS implementations that don't closely follow this description may leak information to an attacker in ways that allow the attacker to distinguish between valid and invalid messages. Although the original vulnerability is about 20 years old, many variants have been proposed and a family of similar exposures persists to this day [13-15] When the vulnerability was first disclosed, the designers of TLS chose to keep the vulnerable encryption modes in service and add countermeasures, which themselves were later found to be incomplete, necessitating the deployment of increasingly complicated countermeasures which are often not implemented properly. Recent research has demonstrated similar Bleichenbach-style vulnerabilities in XML encryption, S/MIME, Java Object Signing and Encryption (JOSE), and other approaches, which persist even within large organizations. Vulnerable hosts relying on RSA encryption key exchanges are exposed to attackers that passively record traffic for later decryption, while hosts using forward secrecy remain vulnerable to a server impersonation or man-in-the-middle version of the attack. Even if an application does not use RSA, and is thus not directly vulnerable to this attack, it may be possible to indirectly attack post-quantum RSA; for this reason, the developers of the ROBOT attack recommended that Ethereum immediately switch to Quantum Blockchains [15]. In addition to rigorously following the often confusing recommendations in RFC 5246, other recommended forms of mitigation include disabling RSA

encryption (many modern TLS connections use Diffie-Hellman ECC keys, and rely on RSA only for signatures).

Since our identity solution implements TAC, we can effectively defend against these attacks. A common feature of Bleichenbach-style attacks is that they typically requires access to TLS error messages, which can be suppressed by the transport layer access control implemented in our authentication appliances. This has the advantage of combining protection from Bleichenbach-style attacks with another form of access control that may already be in use on the network; further, this approach is significantly less complicated to implement that the RFC 5246 recommended mitigations, and would not require the use of quantum blockchains for Ethereum security.

CONCLUSIONS

We have investigated security considerations for an Ethereum blockchain hosting a smart contract for a distributed energy management application. First, we simulated a microgrid with ten buildings in the northeast U.S., and studied both energy balance and transaction rates. We demonstrated that typical energy management can be disrupted if the identity of the smart meters are compromised. Then we investigated a first packet authentication and transport access control scheme for improved authentication and identity management in the smart contract. With negligible performance impact, we are able to successfully mitigate attacks which target specific ports (i.e. Bleichenbach-style attacks) as well as port scans and DDoS attacks. Our identity-based security approach overcomes the limitations of current ERC 725 based identity management, and eliminates the need to migrate Ethereum to a quantum blockchain to achieve improved security.

ACKNOWLEGMENT

We gratefully acknowledge the support of Marist College and the New York State Cloud Computing and Analytic Center (CCAC), as well as support from the National Science Foundation under CC*DNI Integration (Area 4): Application Aware Software-Defined Networks for Secure Cloud Services (SecureCloud) Award #1541384.

REFERENCES

[1] A. Nordrum, “Wall Street firms to move trillions to blockchain in 2018”, IEEE Spectrum, October 2017, https://spectrum.ieee.org/telecom/internet/wall-street-firms-to-move-trillions-to-blockchains-in-2018 (last accessed November 20, 2017)hg

[2] K. Lewis, “Blockchain: four use cases transforming business”, IBM Internet of Things blog, May 2017 https://www.ibm.com/blogs/internet-of-things/iot-blockchain-use-cases/ (last accessed November 20, 2017)

[3] K. Lotay and C. DeCusatis, “Using blockchain technology to digitize supply chain systems”, Proc. National Conference on Undergraduate Research, Atlanta, GA, Nov. 3-5, 2017

[4] M. Peck, “Blockchains: how they work”, IEEE Spectrum, October 2017, https://spectrum.ieee.org/computing/networks/blockchains-how-they-work-and-why-theyll-change-the-world (last accessed November 20, 2017)

[5] Cisco Institution, “Cisco 2017 annual cybersecurity report,” Cisco, Tech. Rep., 2017.

[6] Mikko Hypponen. and Tomi Tuominen., “F-Secure 2017 State of Cybersecurity report,” F-Secure, Tech. Rep., 2017.

[7] M. Peck and D. Wagman, “Blockchains allow rooftop solar energy trading”, IEEE Spectrum, October 2017 https://spectrum.ieee.org/computing/networks/blockchains-will-allow-rooftop-solar-energy-trading-for-fun-and-profit (last accessed November 20, 2017); see also O. Saadh, “U.S. Microgrids 2016 Forecast”, GTM Research Report (August 2016); full report available from https://www.greentechmedia.com/research/report/us-microgrids-2016#gs.ItFERzA and summary by E. Wood available from https://microgridknowledge.com/us-microgrid-market-gtm/ (last accessed June 10, 2018)

[8] A. Rutkin, “Blockchain based microgrid gives energy to consumers in New York”, New Scientist, March 2016 https://www.newscientist.com/article/2079334-blockchain-based-microgrid-gives-energy-to-consumers-in-new-york/ (last accessed April 25, 2018)

[9] Press Release, “Tennet unlocks distributed flexibility via blockchain”, March 2017 https://www.tennet.eu/news/detail/tennet-unlocks-distributed-flexibility-via-blockchain/ (last accessed April 25, 2018)

[10] R. Schlesinger, “Ethereum based solar energy platform to energy damaged Puerto Rico”, BitOnline, December 2017 https://www.bitsonline.com/blockchain-solar-puerto-rico/ (last accessed April 25, 2018)

[11] S. Nakamoto, “Bitcoin: a peer to peer electronic cash system”, Oct. 31, 2008 http://nakamotoinstitute.org/bitcoin/. http://bitcoin.org/bitcoin.pdf. https://github.com/saivann/bitcoinwhitepaper. (last accessed Sept. 20, 2017)

[12] Ethereum white paper, “A next generation smart contract and decentralized application platform”, https://github.com/ethereum/wiki/wiki/White-Paper (last accessed September 20, 2017)

[13] D. Bleichenbacher, “Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS#1”, Proc. Crypto ’98, LNCS 1492 (H. Krawczyk, editor), pp. 1-12 (Springer-Verlag, Berlin), 1998 http://archiv.infsec.ethz.ch/education/fs08/secsem/bleichenbacher98.pdf

[14] Decentralized Identity Foundation, http://identity.foundation/ (last accessed April 24, 2018)

[15] P. Braendgaard, “Different approaches to Ethereum identity standards”, uport blog https://medium.com/uport/different-approaches-to-ethereum-identity-standards-a09488347c87 (last accessed April 24, 2018)

[16] F. Vogelsteller, ERC 725 standard proposal, https://github.com/ethereum/EIPs/issues/725

[17] ERC 20 Token Standard, https://theethereum.wiki/w/index.php/ERC20_Token_Standard (last accessed April 24, 2018)

[18] C. DeCusatis, M. Zimmerman, and A. Sager, “Identity based network security for commercial Blockchain services“, Proc. 8th annual IEEE Computing and Communications Workshop and Conference, Las Vegas, NV (January 8-10, 2018)

[19] Github site for Ethereum project, https://github.com/kslotay/microgrid-incipere (last accessed April 24, 2018)

[20] C. DeCusatis, “The NSF SecureCloud Project: Cybersecurity for Autonomic, Zero Trust Cloud Data Centers”, NSF Internet2 Cybersecurity Transition to Practice Workshop, New York, NY (April 17, 2018)

[21] C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, “Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication”, Proc. IEEE International Conference on Smart Cloud (SmartCloud 2016), New York, NY (Nov. 18-20, 2016)

[22] TLS protocol standard version 1.2 https://tools.ietf.org/html/rfc5246 (last accessed April 24, 2018)