A novel Paradigm for Access Control Trust in IoT Applications: A Distributed Cross-Communication Approach

Muath A. Obaidat, Senior, Member IEEE
Computer Science Department
City University of New York
New York, New York 10019, USA
muobaidat@ccny.cuny.edu

Joseph Brown
Computer Science Department
City University of New York
New York, New York 10019, USA
joseph.brown1@jjay.cuny.edu

Abdullah Al Hayajneh
Professional Security Studies
New Jersey City University
Jersey City, NJ 07305, USA
aalhayajneh@njcu.edu

***Abstract*—How to best secure communications between IoT devices remains a challenging research area. However, the heterodox nature of individual IoT devices and its limited capabilities demand a more centralized, authority-derivative nature of SSL/TLS for the IoT ecosystem. While a number of experimental protocols have been proposed, many of these techniques are either derivative of SSL/TLS on some layer, or lack either enough ubiquity to be widely applied to a live ecosystem, or verifiable results. This paper proposes a distributed system for securing end-to-end cross-device communications; Access Control Trust A Distributed Cross-Communication scheme (ACT-DCC) using a localized web of self-determinate nodes in a network. As opposed to distributed systems, which typically rely on blockchain or centrally signed certificates, the proposed scheme instead operates on an iterative, decentralized method, which combines pathfinding techniques alongside an algorithmic interpretation of hash-chaining and ledger-based storage. An initial network device carries an original key, which is naturally spread to additional connected devices on the same network in a distributed, web-like fashion. As more devices join the distributed network, the strength of the network grows exponentially through the volume of edge devices available in the web. The proposed scheme shows significant improvement as compared to other cross-device communication schemes, and encourages better-individualized decentralization, and less locally stored data.**

Keywords*—— internet of things, IoT, trust, communication, security, authentication, Blockchain

I. Introduction

IoT faces constant cybersecurity challenges, and remains an open area for research. While many have proposed solutions for various open-ended concerns, discussion has largely shifted to focus on a select handful of architectures as opposed to wider, yet more specific, solutions. Among other cybersecurity issues in IoT, recent analyses have focused specifically on issues of authentication, communication, and device identification. Many of these areas fall under the label of information security; in particular they are concerned with a few overlapping tenets - availability, confidentiality, and integrity (ACI) [24]. As opposed to regular internet-connected devices, IoT devices face unique challenges because of few inherent factors; first and most prominently, the heterogeneity of the ecosystem, stemming from lack of standardization [24,25]. Second, IoT devices suffer from a lack of resources - both minimal local storage, as well as a lack of processing power. This gap in device power has led to distinct translation issues with typically recognized security schemes and concepts used commonly with other devices. The lack of similar resources and standardized environments means that such methodologies cannot generally be extrapolated identically. Research has been done for developing specific security frameworks for IoT devices in lieu of this.

While platform security for these devices is widely discussed, inter-device communication has not received as much scrutiny; in part because of the aforementioned issues with a lack of standardization between devices. However, this is not a subject, which can be ignored - as these devices still exist on the same networks and often communicate; it is unrealistic to think that end-to-end security for IoT devices is a topic, which can be relegated to simple fixes. Within the realm of information security, cross-device communication is a topic that embodies the facets of ACI. Unlike platform connectivity, cross-device communications face further unique challenges; decentralization, and lack of distinct end-point security on devices.

This study proposes Access Control Trust a Distributed Cross-Communication (ACT-DCC) scheme for strengthening cross-device security for IoT devices without bloating needs for local storage or processing power, and for maintaining an efficient speed and deployability. The purpose of the methodology is to assist in productively ensuring the tenets of information security, especially confidentiality and integrity of actions between nodes. The scheme does not use blockchain explicitly, but takes into account similar ledger-based storage and verification techniques to achieve similar results.

The rest of this paper is organized as follows: Section II provides background and a brief overview of state of the art in the field, Section III illustrates the dynamics of the proposed scheme, Section IV discusses results from a prototype implementation of the scheme, and Section V concludes the paper.

II. RELATED WORK AND BACKGROUND

The IoT can be described as an interrelated system of computing devices provided with unique identifiers, and the autonomous ability to transfer data over a network [12]. The following features are considered crucial to IoT structures; self-organized and fully distributed architectures, possibilities for

ISBN 978-3-903176-42-3© 2021 IFIP

heterogeneous networks integrated into a single device, low latency networking, and adequate connectivity to the cloud [12]. Scalability is another important facet within security frameworks [6]. Self-organization can be described as the accepting, processing, and distributing of information between devices autonomously. As this is an intrinsic trait of IoT functionality, risk must be mitigated. Securing device-to-device self-organization is an apropos measure [3]. Information security within IoT is more vital than on other systems because of the aforementioned frequency, autonomy, and spread of data within such networks [1].

Within IoT stems, information security can be broken down into sub-sections separated per layer; information application security (such as direct application data) falls under the application layer, while processing security (such as middleware and cloud computing) fall under the information processing layer. As the processing layer acts as a middleman between networking and application, its security is of utmost priority for ensuring authenticity and confidentiality [11, 24, 25]. It should be noted that security is further complicated depending on what layer is secured - for example, communications on the application layer differ from the network layer. Authors in [19] proposed solutions, but are not concerned with application-layer security, despite this being the most vulnerable layer to cyber-attacks; this is a critical distinction, which this study addresses.

Securing communications for IoT services between devices is pivotal for overall network security. Often IoT devices are the weakest link in wider systems and this becomes more apparent when devices cross-communicate [14]. IoT systems face a number of information security threats, including vulnerabilities to eavesdropping, replay, man-in-the-middle, and certificate manipulation among others [16].

Despite the need for secure communication, device authentication remains vague within the IoT. One commonality in all studies is the need for general secured communication, not just one directional [4]; one directional security does not account for practical usages of most IoT systems, which this study also focuses on addressing. Most secure authentication schemes focus on three traits; network assumptions, communication sessions, and usership [16]. This is in part because typical authentication issues are compounded within IoT environments because of the need for continuous authentication. This is due to the frequency of autonomous cross-device transactions [19].

Across many studies and practical usages, it is agreed upon that IoT objects must be uniquely identifiable and have autonomous authentication [17]. New methods of identification for device communication architectures need establishing [1]. Whether or not IPV6 alone is sufficient for device, identification has been debated - because of both the sheer scope of identifiers required, as well as the scale of usage the protocol must adhere to. Some studies such as [17] proposed a virtual initialization similar to IPV6 for virtual identities separate from the network layer. This evaluation was sufficient, especially during the authentication stage, in its display of a foundation for virtual identities for layers above the network layer. Other studies such as [20] also advocate for identification in the form of “digital watermarks”; essentially a digital signature for data, however, certificates, which encourage centralization, can become a logistical issue, and determinate centralization is not fully compatible with practical IoT system usage.

Other research also suggests that the heterodoxy of IoT usages creates caveats in authentication, and thus there is a need for mutually authorized, interoperable protocols independent of applications’ needs; this is an incredibly important note to consider in context of this study. In the past this has been an issue as studies show that linkage between heterogeneous devices may cause unknown problems, a consideration with autonomous cross-device communication [19].

The most deployable communication schemes abide by lightweight architectures. There are common issues that still apply to even the most successful research proposals; these include reused logins for devices, and vulnerabilities to node capture and impersonation, bypassing network gateways, as well as replay and forgery attacks, and offline password cracking/brute-forcing. This is compounded by the fact that devices often work autonomously [16].

Building trust in networks between systems is an important part of both centralized access control and proliferating secure communications between nodes in a wider system. There are three main ways of building trust within a system; either establishing trust using a centralized root platform within a network, establishing trust using a singular centralized entity individually connected to nodes, or decentralizing a network from an initial trust node [7]. Some level of centralization for building trust is likely unavoidable, but debate over the usage of external platforms, clashes with tenets of IoT inherent decentralization and limited resources [5].

There are drawbacks of centralized systems such as failure of a centralized authority, which are especially seen on scalable IoT networks. Because of the weaknesses native to IoT devices as well as their exponential size, risks are compounded for centralized systems on such networks. Distributed trust schemes can be used to overcome such issues; (e.g. locally centralized networks) circumvent some issues of centralized networks by allowing a hybrid of centralized and distributed systems. However, some level of risk still exists in these hybrid centralized systems; failures of centralized systems still apply to all nodes within a system [7].

In [18] more prevalent security threats in current device-to-device communications for IoT than other facets of the system mentioned, yet such threats have not received much holistic attention. Security on networks for securing transmissions between devices is rarely sufficient [20]. Connections to cloud networks are often more holistic, but often are representative of separate functionalities, and therefore do not always overlap. Certain forms of IoT devices do not need security for their transactions because the data is not confidential; they ignore that IoT devices can be used as entry points into a wider network [15]. Thus, it is better to assume all communications should have some form of security. Among other challenges and caveats within IoT ecosystems comes from the actual usage of IoT systems. For example, the functionality - and thus security - of IoT devices in smart cities differs significantly from the usage of IoT devices within a business, or within a small home [3]. This is not to say that there are not aspects of these niche methodologies, which cannot be extrapolated, in a wider context. The authors in [8] explicitly say they are not created with extrapolation to device-to-device communication in mind, [13] reiterate this point regarding functionality being intrinsic to selecting a form of security. Popular lightweight end-to-end communication frameworks for IoT devices include modified versions of IPsec methodologies. Admittedly, however, even with these popularly cited papers, there is debate over how to best secure IoT individually at each level; device, network, and system. The ever-changing IoT environment is in part what has led to a lack of ubiquity among proposed frameworks [6]. SSL/TLS is often used as a mean of authentication; since most communication happens with remote platforms. However, because of limited storage, there are limitations to local trust-stores, especially on non-standardized devices. One protocol built off SSL is MQTT, which is the most popular protocol for IoT devices for publish/subscribe-based protocols. This protocol uses a remote platform. Other such methodologies include Kerberos, which is an access control model for tokenizing clients [21]. IoTVerif has been proposed as a method of further building off MQTT and SSL/TLS security for device communication [10]. These protocols have shown promising results, but not without adding additional centralization. This centralization is one that has yet to be fully resolved within the field.

Alternatives that are not centralized in some manner typically rely on blockchain for decentralization, which has its own challenges [10]. Blockchain as a proposed model for IoT has issues with scaling, due to network system scalability, blockchain may not be a universal solution to all of IoT challenges [3].

III. DYNAMICS OF ACT-DCC SCHEME

It is important to note that the ACT-DCC draws upon concepts that have prior existed within the public sphere having been verified for practical usage elsewhere, but either not applied to IoT systems or in conjunction with other referenced concepts. This is done for two reasons; firstly, to ensure that the system does not need bottom-up scrutiny due to reliance on existing, prior secured derivative concepts, and secondly to show that IoT systems can be secured even with given existing technologies.

It is largely based on the concept of propagation; creating interconnected trust networks through organic connections, but then using this pre-existing fostered web of connections between individual devices to reinforce permissions on the network itself. This is a contrast to blockchain, for example, in which all nodes must convene with all other nodes. This allows for nodes participant in the system to retain activity and connections they normally would anyway (thus also reducing ledgers significantly), but still participate in the overall reinforcement of the permissioned network. Propagation is not a "new" concept per-se, in fact it is somewhat integral to the understanding of networking, but in discussions of security, it has been largely exchanged for discussions of blockchain, despite incompatibilities stemming from such. This scheme is largely based on propagation as its primary methodology of inter-device trust building.

To explain the devised scheme, we have separated the explanation into two categories; logistical and technical design. The logistical design of this scheme adheres to the tenets of *emergent coordination*, the structure of which can be seen in Figure 1. Emergent coordination is effectively propagation through otherwise organic connections.

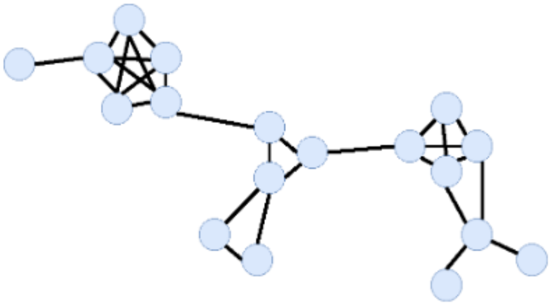

Fig. 1. An example of network nodes propagated through emergent coordination; shows how not all nodes communicate with each other, but organically create an interconnected web between normal communications

It is important to note that this scheme is not a replacement for physical layer connection. Instead, it creates a virtualized series of decentralized device-to-device communications [17], knowing that virtual identification is sufficient in propagating trust-based security within an IoT system.

The overall concept of the scheme facilitates exponentially building trust organically between devices using analog logic without third parties, cloud platforms, or AI/ML. A network based on emergent coordination is decentralized, but not in the common manner. Instead of a network where all nodes are connected, or a standardized centralized network of connections through hierarchy, nodes are semi-centralized in naturally occurring local clusters that exist without singular centralization. Nodes are considered decentralized due to a lack of singular centralized authority. The propagation of this network works two folds.

The virtual network is not meant to replace typical IP routing within a wider network, rather instantiate a virtualization for data-based communications in order to bolster security within such framing. Therefore, this system is compatible with IPv4 or IPv6 routing without any need for low-level changes, physical or otherwise, to IoT devices. It is worth noting the limitation with local storage on an IoT device, [22] estimates that low-power devices have between 10 and 100 kb of additional memory for data retention/processing. So, the goal is to minimize tracking information without having to store a bloated ledger.

To begin, we must start with how the technical design works, however. This is sorted in the following phases, based on behavior; *initialization, connection (retention and/or propagation)* and *transaction.* Each of these behaviors are broken down in details below, in order. These behaviors can be thought of like a state machine, pictured below in Fig. 2. *Initialization* is where the virtual identity is created (how the node will be identified by and to others); it is created upon first network initialization one time, and is only triggered again when and if the propagation fails. *Connection* is what triggers the state machine to actually begin its loop, which is symbolized by an incoming connection from another node (whether unknown or known). After that, what happens depends on conditions within the state machine loop. If the IP is known, it will jump to the *transaction phase.* If the IP is unknown, then the next step depends if the ledger is empty or not. However, if the ledger is empty, it will jump to the *retention* phase. If the ledger is not empty, it will attempt *propagation.* If *propagation* fails, it will return to *retention*. Afterwards, the *transaction* phase will begin.

Initialization is the starting phase, and is where the virtual identity of a node is created. This phase also marks the generation of a clean connection ledger. A virtual identity is represented as a derivative of the IP address of a device, in order to guarantee a unique baseline for an identifier of a given device without repetition on any given network. A hash identity is created by taking an IP address as an input for a SHA256 hash. A SHA256 hash of a randomized string of equivalent length to the IP is generated. The SHA256 hashes are XORed together, and then stored locally; this is referred to as the node's "identifier". As SHA256 produces a consistent length of 64 characters and we are assuming UTF-8 encoding, this means the resulting identifier is 64 bytes.

1. IP_1 = SHA256(IP)
2. $Base_1$ = Random String(Len(IP))
3. Pad_1 = SHA256($Base_1$)
4. ID = $IP_1 \veebar \texttt{Pad}_1$

The next phase is *connection*. This happens when two devices attempt device-to-device communication. Ledgers exist on the side of communicating nodes, which retain indexed ID information. If a connection can be found in the index, it can proceed to the *transaction* phase. For the sake of explanation, we will assume the ledgers of both nodes are empty to discuss a first time connection. As such, before continuing we will first discuss the *retention* phase. The *retention* phase happens conditionally if one of two requirements are met; either the ledger is empty, or a transaction cannot be verified through *propagation* which is further detailed once we return to the *connection* phase. A signatory proof must be distributed between parties; this can be looked at akin to a user registration system. This transmission happens through a Fernet Cipher process. Fernet is used instead of other algorithms for speed and because of lack of resources available. Traditional asymmetric encryption methods by comparison are a burden on low-resource devices,

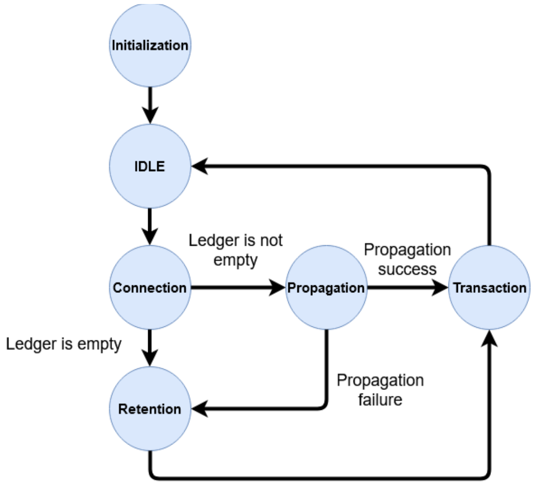

Fig. 2. State machine representation of node behavior; it is essentially a state machine with simple conditional behaviors within a loop, which is consistent post-initialization

A randomized one-time value of 32 bytes is generated, temporarily retained, and sent to the connecting node. The other node receives this, and temporarily stores it as well. Using the randomized value as the key, the node then uses this, encrypted in url-safe base64, as the Fernet key for encrypting the next transfer sent over. Using the Fernet key as a seed for a pseudo-random number generator, a randomized value is generated then hashed with SHA256, then XORed with the locally stored identifier. Encrypted with the Fernet key, this value is sent to the other node. Upon receiving it, the other node uses the temporarily retained value to encrypt with base64 and then use as a Fernet key to verify the message, then uses it as a seed to generate the equivalent SHA256 pad. The message is unpadded by XORing the message with the pad again.

1. ${}_cValue_1$ = Rand(32 bytes)
    a. Random value of 32 bytes generated by node 1 ("client")
2. $sValue_2$ = Base64(${}_cValue_1$)
    a. Base 64 value of received value from client taken by node 2 ("server")
3. ${}_sF_1$ = Fernet(${}_sValue_2$)
    a. Fernet function started with key of base 64 value by node 2
4. ${}_sPad$ = SHA256(Rand(seed=${}_sValue_2$))
    a. Pad created based on seed from base 64 value
5. ${}_sID_2 = {}_sID_1 \veebar {}_sPad$
    a. The actual virtual ID is padded to create ID_2
6. ${}_sValue_3 = {}_sF_1({}_sID_2)$
    a. The ID is ran through the Fernet cipher
7. ${}_cValue_2$ = Base64(${}_cValue_1$)
    a. Original node 1 can recreate node 2's $Value_2$ with its original $Value_1$
8. ${}_cF_1$ = Fernet(${}_cValue_2$)
    a. Fernet cipher is created on node 1, identical to server
9. ${}_cValue_3 = {}_cF_1({}_sValue_3)$
    a. Received value from node 2 on node 1 is put back through fernet cipehr to decpde
10. ${}_cPad$ = SHA256(Rand(seed=${}_cValue_2$))
    a. Pad is recreated with node 1's $Value_2$
11. ${}_sID_1 = {}_cValue_3 \veebar {}_cPad$
    a. Node 2's ID can be found through XORing the Pad out of the ID

The process is then done again vice-versa, so both nodes will retain the identifier of each other. These identifiers are stored locally with the searchable index of the connector's IP. Assuming IPv6, this would be 16 bytes (address length) + 64 bytes (identifier length), or 80 bytes total, or for IPv4, 4 bytes + 64 bytes respectively, or 68 bytes total. If we are to assume on the low end of storage with 5 kb of storage per IoT device (assuming 5 kb for other needed memory), this would allow 62 IPv6 signatories, or 73 IPv4 signatories.

If the *retention* phase has been completed, the state machine moves to the *transaction* phase. In the case, a ledger is not empty; the *propagation* phase takes place instead of the *retention* phase. This is where the bulk of the scheme's unique methodology takes place. During propagation, the IP to validate and the identifier is received, and passed through the ledger retained in each node. As each node evaluates the authenticity of the identifier, it recursively passes it through its own ledger.

If a node has already received the identifier once during this propagation and receives another request to validate it, it will not, preventing duplication. Depending on validation, each node returns an inverse bool; 0 if the index is not found or if it matches stored records, or 1 if the stored record does not match. The collective indexing of all Boolean values is returned bottom up to the original node. The *transaction* phase is where the propagation is transmuted into a usable form of structuring access control for the connection itself, depending on what the connected transaction entails. A visual example of access control breakdown can be seen in Table I. Given propagation, the returned values are compared against the total N nodes.

At this top level, the evaluation is based on comparison; the number of 0s returned is compared to the number of 1s returned. The 1s are given priority, meaning the comparison is 0s > 1s, not 0s >= 1s or otherwise. If 0s > 1s is evaluated to be false, then the propagation is considered false. The failure then leads back to the *retention* phase. If it is evaluated to be true, then it moves to the *transaction* phase involved in the propagation process. From this, thresholds proportional to the total N are used to structure access control For example, using the above table's logic, if the percentage threshold of the first level of access control, N_1, is 25%, then at least 1/4th of propagated nodes must recognize the node, but not more than 50%, which would instead put the node in access control level N_2. By default, or if there is less than the minimum threshold to be within the role of N_1 , then the access control level is considered N_0.

TABLE I. An example breakdown of N-based access control

ACL	Nx Threshold	% Threshold	Permissions
0	$x < N_1$	0%	Send/receive
1	$N_1 > x > N_0$	25%	Send/receive, read
2	$N_2 > x > N_1$	50%	Send receive, read, write
3	$N_3 > x > N_2$	75%	Send, receive, read, write, audit
4	$N_3 < x$	100%	Administration

For example, if only retention has occurred, then the access control level is considered to be 0. Results from propagation are iterated and compared against thresholds set by an administrator with each iteration above 0 easing restrictions. As in line with best practices described in [12], this architecture is considered restrictive rather than permissive, meaning that access control is extremely restrictive from level 0 and as we move upward with each level based on the proportional threshold to the total N, we remove restrictions, rather than each level adding permissions. After a transaction has occurred, the node returns to its idle state.

IV. DISCUSSION AND RESULTS

This distributed system primarily is compared to blockchain-based trust systems, which are among the most prominently proposed for IoT networks. Given this, it is important to compare the benefits of an *emergent coordination* network over a *consensus coordination* network, as traditionally used by blockchain.

Firstly, and perhaps most importantly, resource usage and scalability. As mentioned in the prior section, the average IoT system has between 10kb to 100kb of memory for retention. Assuming usage of half of that storage by other tasks, this would leave us with 5kb to 50kb of retention memory. If we are to assume each signature is equal to the signature described in the prior section (68 bytes for IPv4 signatures or 80 bytes for IPv6 indexing), then the maximum ledger size - solely for signatures, not including other transactions (which blockchain would need to store, depending on structure) - would be 73 or 62 respectively. This max limit would apply to the whole ecosystem, meaning that is the maximum amount of connections on a low-end system (assuming 5kb as the low end). By comparison, if we are assuming a system based on emergent coordination, since there is no universal consensus and nodes are organically propagated, the scaling is infinite, and resources are not hogged by connections unrelated to the direct cluster each node is involved in. Especially when considering heterodoxy of storage in devices, by comparison, the growth of a blockchain device cluster is limited to the resources available in the smallest devices. Blockchain, and other similarly proposed alternative solutions, typically rely on resource-expensive and time-consuming methodologies, such as asymmetric encryption.

This process is built around access control, and for its cryptographic facets uses fast, non-intensive protocols such as Fernet and SHA256, while primarily using low resource, and quick symmetric encryption operations such as XOR and AND.

This scheme retains decentralization, while adhering to the tenets of IoT (i.e. self-organization), without involving authorities or centralized entities. Because of a lack of local resources, centralized trust either relies on local storage and/or processing power - especially for asymmetric encryption-based validation. This is not only unsustainable on IoT, but the assignment of centrally signed signatures, which need to be centrally verified during each transaction, is not probable given IoT network sizes. This is another reason why blockchain is typically heralded for IoT typically; this scheme shares these principles with blockchain, but modifies them to fix scalability and resource management issues as mentioned above. Furthermore, beyond the need for the administration to set the relative N thresholds related to the iterative access control, this scheme is self-propagated, and thus does not require any form of centralized elements - hybrid or otherwise. This allows an organic system to dictate access control features to its elements without worry of central elements failing.

It is worth mentioning that what allows the facilitation of this process is the low resources involved on IoT systems to begin with, alongside the current norms of network communication speeds. As transactions between nodes will inherently never be greater than their working memory, thus never topping the respective 10kb to 100kb figures. As such, we can assume that this amount will be the basis of any conducted transactions between nodes, with the average transaction being fractional of that maximal amount. With that said, the average internet speed in the United States as of 2017 was 18.7 Mb/s [27].

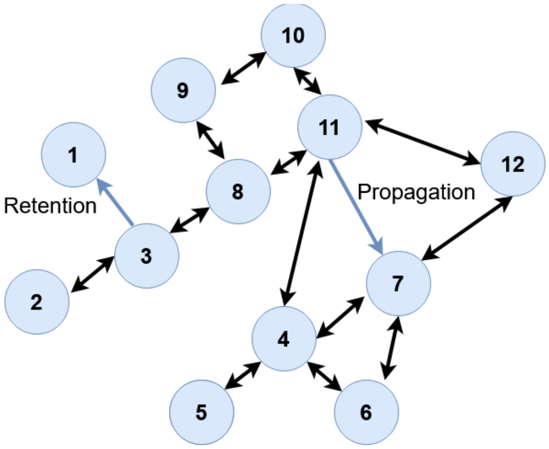

Fig. 3. Test bed visualization. Retention: Nodes 3 and 1. Propagation: Nodes 11 and 7.

This means that we can assume that even the maximal transaction would take fractional milliseconds. Thus, the propagation throughout this network not only logistically rivals blockchain because of not needing synchronization, but can process repeated and frequent network-wide propagation transactions quickly.

Retention tested how long the phase would take between nodes, while propagation tested how long propagation would take within a moderately sized local area network. In Fig. 3 above is a visualization of nodes within the test bed, along with which nodes were tested.

Tables II and III show the speed associated with each phase, separated into categories based on the actions taken for further clarity.

TABLE II. Time needed for each action within the retention phase to occur.

Retention Phase Action	Time (Seconds)
Connection Established; Generate & Send Data/Key	0.06938
Receive, Decrypt & Index; Generate & Send own Data/Key;	0.07625
Receive, Decrypt & Index from other Node	0.07214

TABLE III. Time needed for propagation phase

Propagation Phase Action	Time (Seconds)
Connection Established; Pass to other indexed nodes (Each Connection)	0.0001829
Return Information to Connected Node (Each Connection)	0.0001241
Propagation Returns to Original Node (Entire Propagation Sequence)	0.0217031

For retention, the connection was made between nodes 3 and 1 as shown in the above Fig. 3. For propagation, the study was made between nodes 7 and 11, with all active nodes participating in the propagation sequence except node 1, which was excluded due to the node only being used to test separate retention.

V. CONCLUSION

This study proposes a novel Access Control Trust a Distributed Cross-Communication (ACT-DCC) scheme for self-organized propagation among nodes within an IoT system. The a scheme is based on emergent coordination. The aim is to create an improved form of self-organized device-to-device access control among participating devices in an IoT ecosystem that could retain decentralization without bloating speed, damaging scalability, or unrealistically hogging resources. we found that a system based on emergent coordination not only allowed scaling better than blockchain, but also was more lightweight in terms of computational cost.

REFERENCES

[1] S. G. Abdukhalilov, “Problems of security networks internet things,” in 2017 International Conference on Information Science and Communications Technologies (ICISCT), Nov. 2017, pp. 1–7, doi: 10.1109/ICISCT.2017.8188588

[2] M. Alaslani, F. Nawab, and B. Shihada, “Blockchain in IoT Systems: End-to-End Delay Evaluation,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8332–8344, Oct. 2019, doi: 10.1109/JIOT.2019.2917226.

[3] Z. Din, D. I. Jambari, M. M. Yusof, and J. Yahaya, “Challenges in Managing Information Systems Security for Internet of Things-enabled Smart Cities,” in 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), Dec. 2019, pp. 1–6, doi: 10.1109/ICRIIS48246.2019.9073661.

[4] P. H. Griffin, “Secure authentication on the Internet of Things,” in SoutheastCon 2017, Mar. 2017, pp. 1–5, doi:10.1109/SECON.2017.7925274.

[5] T.-M. Grønli and G. Ginea, “Digital Identities for Internet of Things Devices,” in 2019 IEEE Conference on e-Learning, e-Management e-Services (IC3e), Nov. 2019, pp. 1–5, doi: 10.1109/IC3e47558.2019.8971779.

[6] M. Irshad, “A Systematic Review of Information Security Frameworks in the Internet of Things (IoT),” in 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Dec. 2016, pp. 1270–1275, doi: 10.1109/HPCC-SmartCity-DSS.2016.0180.

[7] H. Kim and E. A. Lee, “Authentication and Authorization for the Internet of Things,” IT Professional, vol. 19, no. 5, pp. 27–33, 2017, doi: 10.1109/MITP.2017.3680960.

[8] C. Lee and A. Fumagalli, “Internet of Things Security - Multilayered Method For End to End Data Communications Over Cellular Networks,” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Apr. 2019, pp. 24–28, doi: 10.1109/WF-IoT.2019.8767227.

[9] Y. Lee, Y. Park, and D. Kim, “Security Threats Analysis and Considerations for Internet of Things,” in 2015 8th International Conference on Security Technology (SecTech), Nov. 2015, pp. 28–30, doi: 10.1109/SecTech.2015.14.

[10] A. Liu, A. Alqazzaz, H. Ming, and B. Dharmalingam, “IoTVerif: Automatic Verification of SSL/TLS Certificate for IoT Applications,” IEEE Access, pp. 1–1, 2019, doi: 10.1109/ACCESS.2019.2961918.

[11] S. Liu, K. Yue, Y. Zhang, H. Yang, L. Liu, and X. Duan, “The Research on IOT Security Architecture and Its Key Technologies,” in 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Oct. 2018, pp. 1277–1280, doi: 10.1109/IAEAC.2018.8577778.

[12] N. Miloslavskaya and A. Tolstoy, “Ensuring Information Security for Internet of Things,” in 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Aug. 2017, pp. 62–69, doi: 10.1109/FiCloud.2017.17.

[13] V. Nagamalla and A. Varanasi, “A review of security frameworks for Internet of Things,” in 2017 International Conference on Information Communication and Embedded Systems (ICICES), Feb. 2017, pp. 1– 7, doi: 10.1109/ICICES.2017.8070757.

[14] S. Narang, T. Nalwa, T. Choudhury, and N. Kashyap, “An efficient method for security measurement in internet of things,” in 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Feb. 2018, pp. 319–323, doi: 10.1109/IC3IoT.2018.8668159.

[15] S. Rajashree, P. Gajkumar Shah, and S. Murali, “Security Model for Internet of Things End Devices,” in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Jul. 2018, pp. 219–221, doi: 10.1109/Cybermatics_2018.2018.00066.

[16] A. K. Sahu, S. Sharma, S. S. Tripathi, and K. N. Singh, “A Study of Authentication Protocols in Internet of Things,” in 2019 International Conference on Information Technology (ICIT), Dec. 2019, pp. 217– 221, doi: 10.1109/ICIT48102.2019.00045.

[17] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, “Identity-based authentication scheme for the Internet of Things,” in 2016 IEEE Symposium on Computers and Communication (ISCC), Jun. 2016, pp. 1109–1111, doi: 10.1109/ISCC.2016.7543884.

[18] V. G. Semin, E. R. Khakimullin, A. S. Kabanov, and A. B. Los,“Problems of information security technology the ‘Internet of Things,’”in2017InternationalConference“Quality Management, Transport and Information Security, Information Technologies” (IT QM IS), Sep. 2017, pp. 110–113, doi: 10.1109/ITMQIS.2017.8085775.

[19] M. Shahzad and M. P. Singh, “Continuous Authentication and Authorization for the Internet of Things,” IEEE Internet Computing, vol. 21, no. 2, pp. 86–90, Mar. 2017, doi: 10.1109/MIC.2017.33.

[20] B. Usmonov, O. Evsutin, A. Iskhakov, A. Shelupanov, A. Iskhakova, and R. Meshcheryakov, “The cybersecurity in development of IoT embedded technologies,” in 2017 International Conference on Information Science and Communications Technologies (ICISCT), Nov. 2017, pp. 1–4, doi: 10.1109/ICISCT.2017.8188589.

[21] A. A. Wardana and R. S. Perdana, “Access Control on Internet of Things based on Publish/Subscribe using Authentication Server and Secure Protocol,” in 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Jul. 2018, pp. 118–123, doi: 10.1109/ICITEED.2018.8534855.

[22] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, “A Survey of IoT Applications in Blockchain Systems,” *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–32, 2020.

[23] D. Belson, Ed., “Q1 2017 State of the Internet,” *Akamai*, May-2017. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf. [Accessed: 30-Sept.-2020.

[24] Muath A. Obaidat, Suhaib Obeidat, Jennifer Holst, Al Hayajneh, A, and Joseph Brown,”A Comprehensive and Systematic Survey on the Internet of Things: Security and Privacy Challenges, Security Frameworks, Enabling Technologies, Threats, Vulnerabilities and Countermeasures.” Computers 2020, 9, 44

[25] Muath A. Obaidat, Joseph Brown Suhaib Obeidat, and Majdi Rawashdeh., “A Hybrid Dynamic Encryption Scheme for Multi-Factor Verification: A Novel Paradigm for Remote Authentication” Sensors Journal. July 2020.