A Blockchain-based Contactless Delivery System for Addressing COVID-19 and Other Pandemics

Pratiksha Mittal*, Austin Walthall†, Pinchen Cui†, Anthony Skjellum◇ and Ujjwal Guin*
*Dept. of Electrical and Computer Engineering, Auburn University
†Dept. of Computer Science and Software Engineering, Auburn University
◇SimCenter & Dept. of Computer Science and Engineering, University of Tennessee at Chattanooga
{pzm0042, acw0032, pinchen, ujjwal.guin}@auburn.edu, Tony-Skjellum@utc.edu

2021 IEEE International Conference on Blockchain (Blockchain) | 978-1-6654-1760-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BLOCKCHAIN53845.2021.00060

***Abstract*—The coronavirus (COVID-19) pandemic has significantly impacted and changed our daily routines. Worldwide, people have had to adapt by undergoing remote work and self-quarantine. This situation has required transforming strategies for various logistics services for a variety of service providers, such as retail stores and restaurants. The concept of contactless delivery has emerged to help prevent the spread of the coronavirus. However, contactless delivery only reduces the direct interaction between the delivery personnel and the customer. In addition to peer-to-peer contact, items still go through insecure interactions between and among the delivery personnel and other unknown third parties. Even if the items are delivered without physical contact, concerns remain about their routes in the supply chain. In this paper, we present a novel blockchain-based framework to enable the traceability of products in the supply chain. This framework records and tracks delivery traces and the medical status of delivery personnel in a privacy-preserved manner, ultimately contributing to COVID-19 prevention and control. We build a Hyperledger Fabric-based blockchain prototype system as our testbed. Several smart contract functions are implemented and evaluated to show the efficiency of the proposed framework. In conjunction with the implementation and evaluation, we also perform comprehensive security and privacy analyses of this framework.**

***Index Terms*—Blockchain, COVID-19, Delivery System, Pandemics.**

I. Introduction

COVID-19 has become one of the most severe pandemics in human history, resulting in an unprecedented disease burden, extraordinary healthcare costs, and deleterious economic impacts worldwide. As of November 3, 2021, the coronavirus has been confirmed in more than 247 million people worldwide. Of these confirmed cases, there have been more than 5 million deaths globally, leading to a mortality rate of approximately 1.6% in the United States [1]. The virulence of the coronavirus may be a result of its considerable stability on surfaces [2], as it can survive on stainless and plastic surfaces for more than seven days [3]. Thus, the disease continues to spread even though significant efforts have been made to quell this pandemic. Despite the recent development of vaccines for the virus, various reports indicate that people need to continue socially distancing and taking proper precautions in the coming months.

The spread of the coronavirus threatens public health and drastically changes peoples' lifestyles. New policies, such as social distancing, telecommuting, and self-quarantines, have been proposed and enforced worldwide to thwart the virus's spread. Although the reduction and regulation of human interaction help control and prevent the virus, it also dramatically influences many business domains, especially businesses with physical stores. For example, the decrease in customer traffic directly impacts the retail store and catering services, leading to job changes and job losses [4]. It is a dilemma to keep people safe while also maintaining regular business operations. As an alternative solution, contactless delivery can partially solve this problem. Contactless delivery provides products to the customer while maintaining fundamental social distancing. It allows one to get daily supplies, such as groceries and medicines, by delivering the items to their doorstep. However, the delivered items go through insecure interactions between and among the delivery personnel and other unknown third parties. As stated above, the stability of the virus may enable it to spread further since it can survive for extended periods of time on fomites such as boxes or bags. We need to further track the delivery information even in the contactless delivery scenarios. This allows the virus transmission route to be further investigated to determine if any confirmed cases have been reported in the system.

Currently, many countries around the world have adopted mobile technologies to alleviate the coronavirus's spread and rely on such technologies to provide information to improve their decision-making on the lockdown exit strategy. In order to promote contact tracing, different countries have taken the lead by requiring their citizens to install surveillance apps [5]–[7]. Unfortunately, people cannot access this information to make risk-informed decisions while they interact with others. Even with the increased demand for home delivery, the delivery personnel is not routinely assessed for coronavirus exposure. Thus, it is essential to develop an infrastructure to allow for risk-informed decision-making while implementing a contactless delivery system. Therefore, we propose a blockchain-based system to record the medical status of the delivery personnel and then trace the infection pathway in the supply chain. By utilizing a contactless delivery system, human-to-human interaction is minimized, and then our proposed framework allows all involved entities to acquire up-to-date information on the risks in the supply chain.

Even though a significant amount of work has been performed to address the COVID-19 pandemic, there is no contactless delivery system that can provide contact tracing in the event of exposure to the coronavirus. Package delivery services are

978-1-6654-1760-0/21/$31.00 ©2021 IEEE
DOI 10.1109/Blockchain53845.2021.00060

transporting packages that have potentially traveled all over many locations. These packages are guaranteed to be sanitized or cleaned, so the delivery personnel is at constant risk of contracting an illness from the package itself. Besides, they may also come across coronavirus contact while traveling a highly contaminated region as a part of delivery. If the delivery personnel is at constant risk, then the recipient of the package shares the same risk. This is why it is crucial for the package recipient to be able to track the location of delivery personnel and take precautions while handling these packages.***As a result, there is an urgent need for developing a contactless delivery system where the location traces of delivery personnel can track anonymously.*** The major contributions of this paper are summarized as follows:

- We propose a novel blockchain-based framework to enable traceability of individuals participating in contactless delivery systems. All delivery personnel from organizations participating in this proposed framework need to update the ledger with a location for delivery. Once a user submits a query, a user (customer) can recover a complete trace from the immutable blockchain ledger while still maintaining the privacy of the individual uploading the information. The user can then determine if there was exposure to the coronavirus and make a risk-informed decision whether or not to handle the delivery. To maintain privacy, it is unnecessary for delivery personnel to disclose his or her personal identity. Instead, a unique identifier will be assigned to each participating individual.
- The proposed infrastructure uses Hyperledger Fabric [8] along with the non-resource intensive consensus algorithm Raft [9] to implement the proposed blockchain framework. We use Hyperledger Fabric's docker containers to generate CouchDB state databases. The importance of using docker containers is that it allows us to isolate chaincodes [8]. The way smart contracts are packaged for deployment is governed by chaincode. By isolating chaincodes, we are able to guarantee the success of concurrent transactions. We provide the latency and throughput at different transaction rates and different batch sizes to evaluate the effectiveness of our proposed approach.
- Our proposed framework not only ensures robust supply chain provenance as it is built upon blockchain technology, but it preserves the privacy of all entities involved in this framework. Since the delivery personnel is required to provide confidential information to the service provider (i.e., a daily health check), our framework guarantees their privacy by associating the delivery personnel with a unique identification number rather than a specific name. The only member of the blockchain that knows the mapping between the delivery personnel and their ID number is the service provider.

The rest of the paper is organized as follows: Prior work is presented in Section II. We introduce our proposed novel blockchain-based framework and the implementation details in Section III. The results of the proposed approach are analyzed in Section IV. Finally, we conclude our paper in Section V.

II. Background

The decentralization of storage in a blockchain infrastructure makes it suitable for a wide variety of applications. A decentralized blockchain is open for anyone to enter and make transactions as well as engage in the consensus process. This decentralized model provides high robustness and durability for the database stored on the blockchain with no single-point failure.

• ***Blockchain Technology:*** In 2008, Satoshi Nakamoto first introduced the concept of blockchain in the seminal Bitcoin paper [10] to solve the double-spending problem in digital currency systems. Bitcoin's success triggered a rapid development and general interest in designing blockchain technology and applying it to different fields. Primarily, the blockchain infrastructure depends on how the consensus mechanisms are performed. There are four fundamental consensus mechanisms in the current blockchain systems, and they are Proof of Work (PoW) [10], Proof of Stake (PoS) [11], Practical Byzantine Fault Tolerance (PBFT) [12], and Delegated Proof of Stake (DPoS) [13]. A few other consensus mechanisms are also used in some blockchain systems, Proof of Elapsed Time (PoET) [14], and Proof of Authority (PoA) [15]. Recently, Hyperledger Fabric [8] has garnered attention in implementing different blockchain-based applications. Hyperledger is a permissioned blockchain that uses the Raft consensus algorithm [9]. Anyone can join the consensus as long as they are a member of the blockchain infrastructure. The Raft consensus algorithm provides feasible performance bottlenecks, making it a preferable candidate for our framework. In addition, it is non-resource intensive, thereby reducing the expense of a transaction fee and increases performance.

• ***Blockchain for Traceability:*** A number of researchers have proposed the use of blockchain technology to tackle problems related to the COVID-19 pandemic. These approaches can be categorized based on tracking technologies [16], [17] and using the tracked data to inform people about COVID-19 risks [17], [18]. In addition, a number of blockchain frameworks have been proposed or implemented in supply chain management, such as the blockchain solution provider TYMLEZ working with the Dutch government to implement a blockchain-based solution to assist with the supply and demand of medical products [19]. One more example is the VeChain network [20]. The VeChain network ensures the credibility and durability of new KN95 masks imported from China while collaborating with production offices and facilities [20]. However, many of the currently proposed systems follow a centralized architecture in which only permissioned users can access the information. One example of this centralized architecture is Singapore's contact tracing solution called TraceTogether. This application employs Bluetooth technology to monitor potential coronavirus exposure between individuals [21], [22]. In addition, the BeepTrace framework [16] uses blockchain technology to provide encrypted and anonymous personal identification, allowing authorities and health care providers to reach out to people who may be at risk of infection due to contact with an infected person.

III. Proposed Delivery Framework

The proposed blockchain-based architecture is created to provide a contactless delivery system for recording delivery traces and the medical status of delivery personnel for various service providers, delivery personnel, and end users/customers.

Service providers, delivery personnel, and customers are members of the proposed blockchain framework. The identity (i.e., address, account, or participant's identity) must be established and maintained in the system by each member (node). To maintain supply chain integrity, the service provider must keep track of the blockchain ledger to observe any changes from order creation to order delivery. This will assist in the event a discrepancy arises [23]. A blockchain transaction has a designated smart contract function, a payload that contains input values to the function call, and is always signed by the submitter. The underlying functionalities that include the actual data storage and management are implemented through smart contracts. Smart contracts are code lines stored and automatically executed on a blockchain when predetermined terms and conditions are satisfied. The blockchain nodes execute these smart contracts by processing transactions submitted by the user.

A. Architecture of the proposed blockchain-based framework

Figure 1 shows our proposed framework that allows customers to send an order to a service provider, who then updates the blockchain with all information regarding the order. Once the service provider's information is uploaded, delivery personnel then updates the blockchain with information regarding the status of the package and delivery locations. There are two functions that are not shown in Figure 1: delete order and edit order. These two functions are only callable by the service provider and will only be invoked when requested by the customer. At any point in the process, the customer can access the blockchain to 1) check the location of the delivery personnel and 2) observe any information about potential coronavirus exposure. In this section, we will focus on the details and implementation of the proposed framework. We implement the proposed framework using the following steps:

• ***1 – Create Order:*** This function allows the calling entity to create and upload a new order to the blockchain. The only entity permitted to call this function is the service provider. To generate the order ID, the service provider could simply concatenate the current date with the customer's last name and pass through this function. In addition, the service provider is also responsible for uploading the health status of the delivery personnel selected to deliver the order. Once the order number is created, and the health status of the delivery personnel is obtained, the *createOrder()* function checks to determine if the caller of this function is a valid entity, based on a predefined set of access policies (Figure 3). If this passes, the order ID, entity uploading the order, the item details and availability, delivery personnel details, and customer details are successfully uploaded to the blockchain.

• ***2 – Update Information:*** The *update()* function is used to update the delivery personnel's information on the blockchain. This function takes two arguments: the delivery personnel's ID number and their current location. The service provider will initiate a blockchain transaction using the *update()* function to update the delivery personnel's ID number and the pickup location. A unique identification number would be used to associate the delivery personnel with the order to preserve the delivery personnel's privacy. For example, the delivery personnel's employee ID number could be used. This way, only the delivery personnel, and their boss would associate that number with the correct individual. While delivering, the delivery personnel use *update()* to provide all locations they stop at on the delivery route. If the delivery personnel becomes sick, then the service provider can trace all stops on the delivery personnel's route and isolate anyone that might have been exposed. The customer is able to monitor this information by querying the blockchain to determine if the order has been exposed to any contamination. Since the ledger is immutable, this function creates a new block containing the updated information and links it to the order block.

• ***3 – Transfer Order:*** Our framework utilizes a function to transfer ownership of the order from one entity to the next. To transfer the order essentially means that the current owner is giving the order to the next person in the delivery chain. The service provider is the initial owner and will then transfer the order to the delivery personnel. Once the delivery personnel is ready to deliver the order, they will transfer the order to the customer. This *transferOrder()* function takes the order ID and the information about the new owner as arguments. Our access policy only permits the service provider and the delivery personnel to use this function. When called, this function triggers the smart contract to change the current owner to the new owner. The *transferOrder()* function provides security for the supply chain by providing backtracking capability to determine where a package may have been lost or stolen. Also, the transfer order functionality provides non-repudiation by providing backtrack capability to show official ownership of the item at each step in the delivery chain.

• ***4 – Confirm Delivery:*** The final step in our framework is receiving confirmation that the order has been delivered. Since our framework is based around a contactless delivery system, it is the customer's responsibility to confirm the completed delivery. To do this, the customer and delivery personnel will utilize the *setDelConfirmation()* function. This function takes the order ID, transfer confirmation (i.e., Yes/No), and delivery location as arguments. When this transaction occurs, the arguments will be compared with their expected order ID, transfer confirmation, and delivery location values. If the information matches, then the delivery can be considered successful, and the status of the transfer will be updated on the blockchain. Otherwise, the data stored in the blockchain allows backtracking to determine where an issue may have arisen. In addition, by confirming the delivery, the customer can not claim the creation of the order to the service provider.

• ***Location Tracking and Health Monitoring:*** To monitor the delivery personnel's location and health status, the customer and the service provider can use the *getTrace()* and *getMedicalStatus()* functions. Both of these functions effectively work the same way and take the same argument: order ID. When called, the function will automatically query the blockchain for whichever function is being used and return either the location information or the delivery personnel's medical status. By recording the location and medical status of the delivery personnel, contact tracing can be utilized to map a potential infection pathway in the delivery system in the event of an infection. Contact tracing would allow the service provider to identify all delivery personnel involved with the same route and remove them from the delivery options. This also allows the

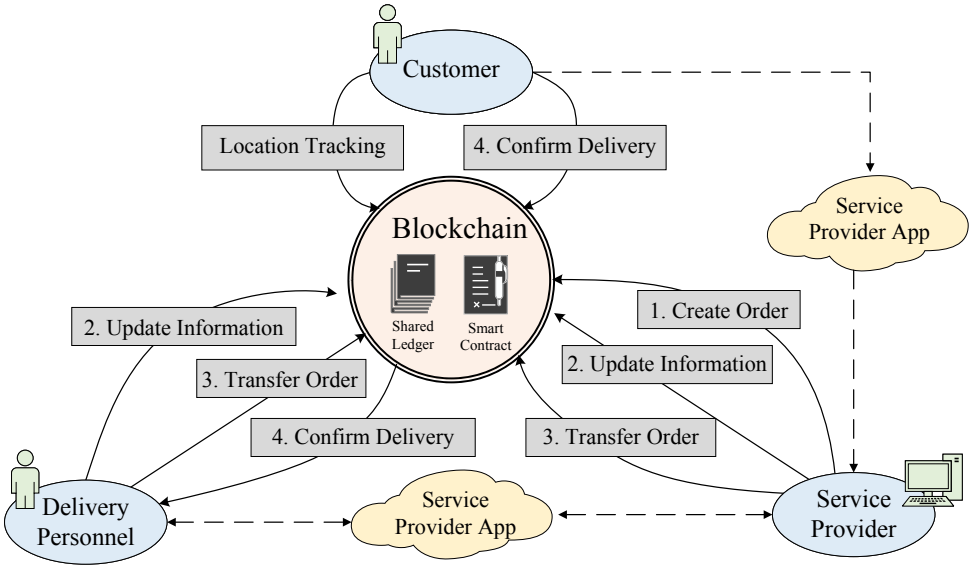

Fig. 1. The proposed approach for the blockchain-based contactless delivery system.

customer to be aware if their package could be contaminated and then take necessary precautions to protect themselves. In addition to customer safety, since the delivery personnel is only associated with an identification number, their privacy is preserved.

• ***Delete Order:*** If the customer places a wrong order or does not need the order anymore, they can request the service provider to delete them. If the service provider agrees, the service provider will call the *delOrder()* function to delete the customer's order. This function takes only the order ID as an argument. The order can only be deleted if delivery personnel has not been assigned for the delivery. If the order has already been transferred to the delivery personnel, there is no way to delete it. By allowing only the service provider access to this function, we ensure that no intermediary entities can delete the order and then steal the package.

• ***Edit Order:*** In case the customer wishes to edit any order information, they can request the service provider to edit the order. If the service provider agrees to the request, they can initiate a transaction using the *editOrder()* function. Only the service provider is permitted to call this function. This function takes the order ID and the customer's information as arguments. If the order has already been transferred to the delivery personnel, there is no way to edit it. Permitting only the service provider access to this request restricts adversaries from maliciously changing information about the order, potentially leading to a lost or stolen delivery.

B. Access Control

To regulate and secure the operations in the blockchain system, we implement an access control policy. Access control allows for control over which entities are permitted to invoke which operations. The core access control policies of our prototype system are depicted in Figure 2. The policies are enforced to give access to the operations; otherwise, the entity will be denied access. The policy $R1$ allows all users to read any of the resources stored in the blockchain. $R2$ grants only the service provider access to modify the order records, including creating, modifying, and deleting the order. $R3$ allows only the service provider to

Rule R1 {
description: ""
participant:
"ANY"
operation: READ
resource:
"com.order.*"
action: ALLOW }

Rule R2 {
description: ""
participant(r):
"com.order.entity"
operation: ALL
resource:
"com.order.orderID"
condition: (r.type ==
"Serviceprovider")
action: ALLOW }

Rule R3 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.orderID"
transaction(t):
"com.order.delete"
condition: (r.type ==
“Serviceprovider”)
action: ALLOW }

Rule R4 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.receiver"
transaction(t):
"com.order.transfer"
condition: (r.type ==
“Deliverypersonnel”
&& r.type ==
“Serviceprovider”)
action: ALLOW }

Rule R5 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.transfervalue"
transaction(t):
"com.order.confirmation"
condition: (r.type ==
“Customer”
&& r.type ==
“Deliverypersonnel”)
action: ALLOW }

Rule R6 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.deliverypersonn
ellocation"
transaction(t):
"com.order.deliverypersonn
elupdate"
condition: (r.type ==
“Deliverypersonnel” &&
r.type == “Serviceprovider”)
action: ALLOW }

Rule R7 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.item"
transaction(t):
"com.order.orderedit"
condition: (r.type ==
“Serviceprovider”)
action: ALLOW }

Rule R8 {
description: ""
participant(r):
"com.order.entity"
operation: UPDATE
resource(d):
"com.order.deliverypersonnel
medicalstatus"
transaction(t):
"com.order.deliverypersonnel
update"
condition: (r.type ==
“Serviceprovider”)
action: ALLOW }

Fig. 2. Access Control Policies.

delete the order, while $R4$ allows the service provider and the delivery personnel to transfer ownership of the order. $R5$ allows the customer and delivery personnel to confirm the delivery of the item. $R6$ allows only the delivery personnel and the service provider to update the delivery personnel's location. $R7$ and $R8$ allow only the service provider to update the order details and

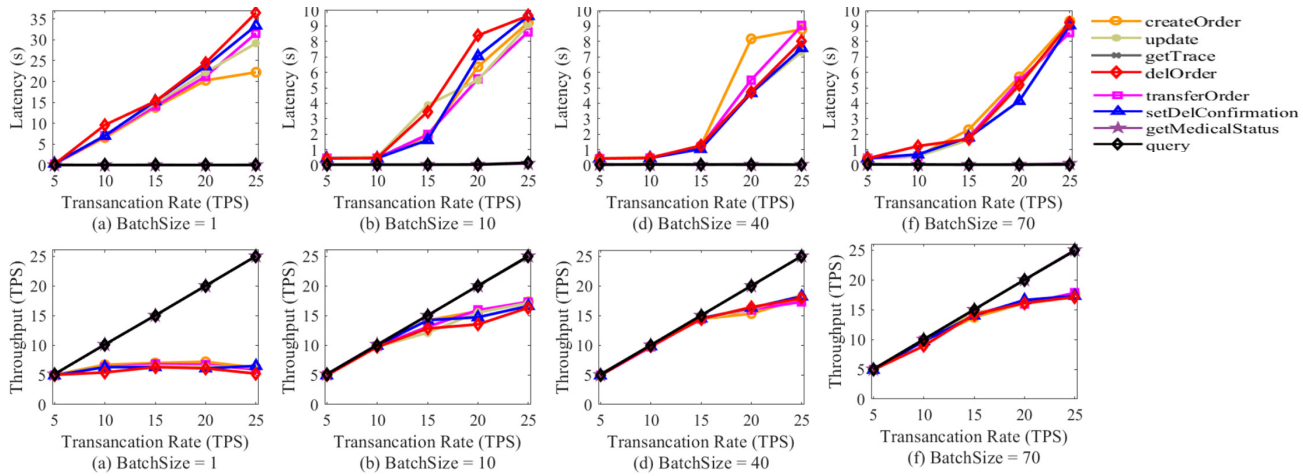

Fig. 3. Latency and Throughput versus transaction rate with different batch sizes.

update the delivery personnel's health status, respectively. While $R6$ and $R8$ apply to the same function, $R8$ is used to restrict the delivery personnel from updating their medical status. We enforce this to ensure that the medical status being updated is legitimate.

C. Usage in the Service Industry

Arguably one of the most desired services during a pandemic is food delivery. Many of the large restaurant chains have implemented some form of delivery service to track the order status. Unfortunately, none of them provide any information related to the COVID exposure of the delivery personnel. Contrary to this, our proposed framework can provide a scalable system to support not only large restaurant chains but also smaller restaurants. Therefore, our proposed blockchain-based framework would provide an indispensable asset for these restaurants and restaurant chains to enable safe delivery during the pandemic time. It would allow them to set up a safe and secure delivery system that they would control with no third-party overseer. The staffs can take food delivery requests from customers and upload the order and destination details to the blockchain using *createOrder()* and *update()* described in Section III-A. These functions will allow the customers to monitor their order status and location information. Once the order has been established, the restaurant would then select an employee to deliver the food. Since most dining establishments already require their employees to undergo daily health screenings, the delivery person's health status needs to be uploaded to the blockchain using the *update()* function. If the delivery personnel has multiple stops for multiple deliveries on their route, the location of each stop will be uploaded into the blockchain using the *update()* function. If they become sick afterward, the restaurant would be able to notify everyone on the delivery route, as well as the employees, that they might have been exposed. Since we are establishing a contact-less system, the delivery person would simply put the food on the customer's doorstep, notify the customer of their delivery, and continue with other deliveries like the traditional ones. When the customer receives the food, they need to confirm that they received it using the *setDelConfirmation()* function. At any point after placing the order, the customer can query the blockchain for information pertaining to the delivery person's health status and location using the *getMedicalStatus()* and *getTrace()* functions, respectively.

IV. Results and Discussion

In this section, we provide quantitative evidence for the success and implementation of our framework. We will also discuss how our framework is protected and secure for all entities involved.

A. Performance Evaluation

To evaluate our proposed blockchain-based framework, we measured throughput and latency. We used three desktop computers for our evaluation environment, each equipped with an eight-core CPU and 16GB RAM. We created ten organizations in a single channel with Hyperledger Fabric 1.4.1 docker containers and CouchDB state databases [24]. We deployed three simulated customers using RAFT on our three machines. We simulated five service providers on machine one to create the orders.

We evaluated our system's latency and throughput by stressing our system with varying transaction rates and different batch sizes. We used batch sizes of 1, 10, 40, and 70 transactions and transaction rates of 5, 10, 15, 20, 25 transactions per second (TPS). Figure 3 shows the latency and throughput of different functions at different transactions with varied batch sizes. The *createOrder()*, *update()*, *transferOrder()*, *setDelconfirmation()* functions perform both read and write operations on blockchain, while the *query()*, *getTrace()*, *getMedicalStatus()* functions are read-only. For the functions that perform both read and write operations, we observe similar throughput and latency behaviors. Alternatively, the functions that use read-only operations have lower latency than the read and write functions. We can observe that the latency increases when the TPS number increases. The trial with a batch size of 1 is the exception to this behavior, as the latency increases linearly for the read and write functions. All other block sizes have a latency of fewer than 9 seconds. For batch sizes greater than 1, the latency dramatically increases in the range between 15 and 20 TPS. This increase in latency can be explained by the number of blocks in which the transactions were packaged and committed. The throughput increased linearly as the transaction arrival rate increased until it flattened out around the saturation

point. The peers became saturated, consuming all of the available CPU and disk I/O allocated to the container. We observe a bottleneck occurring in throughput when the TPS reaches roughly 25.

B. Security and Reliability of the Framework

This section presents the attack analysis of the proposed framework, where all entities involved are permitted to read the data, but only certain entities have the write permission. Even with permissioned access, one could still make the argument for two different concerns for our framework: illegitimate medical test results and illegitimate location information.

• ***Illegitimate Medical Test Result:*** An illegitimate medical result can occur if an employee accidentally or intentionally provides an incorrect COVID test result to the service provider. To address this concern, the test needs to be official and verified by an authentic organization, such as a hospital or COVID test center. Once the results have been verified, the service provider can upload the test result to the blockchain. In addition, the medical test results could be verified by approved officials by implementing a separate blockchain function solely for this purpose. If an incorrect medical test is uploaded to the blockchain, we can implement additional functions to delete or modify the details. We would implement a new access policy rule to allow only trusted entities access to these functions.

• ***Illegitimate Location Information:*** This attack occurs when a permissioned user has updated the wrong location information either by mistake or on purpose. To account for this possibility, we implement location checking in our blockchain-based framework. When an order is either (i) transferred from owner to owner or (ii) delivered to the customer, we require a transaction updating the current location of the transfer. If a service provider is transferring the order to a delivery person, then the service provider uploads their current location to the blockchain. This should match the current expected location of the order to prevent an invalid location update. When the delivery personnel delivers the package, they must also update their current location, which will be cross-checked with the expected location. Upon reception of the order, the customer will finally update their current location, which will then be compared with the (i) initial order delivery address and (ii) the location that the delivery personnel uploaded for the order transfer. If there are any discrepancies in location, it is simple to check the blockchain records to determine where the issue arose.

V. Conclusion

In this paper, we have presented a blockchain-based framework to provide a secure and safe contactless delivery system for COVID-19 and other pandemics. For each order created and delivered in the framework, one could track delivery personnel's medical test status and a trace of travel history of delivery personnel to different locations. All the service providers, delivery personnel, and end-users or customers could benefit from the framework since it helps the customer to have a contactless delivery system. We performed a comprehensive security analysis for this framework to ensure that it is secure and reliable. Additional research is needed to explore the tracing in case of wrong deliveries and order cancellations.

Acknowledgment

This work was supported by the National Science Foundation under grant number CNS-1755733.

References

[1] COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU), https://coronavirus.jhu.edu/map.html.

[2] R. Wathore, A. Gupta, H. Bherwani, and N. Labhasetwar, "Understanding air and water borne transmission and survival of coronavirus: Insights and way forward for sars-cov-2," *Science of the total environment*, vol. 749, p. 141486, 2020.

[3] N. Van Doremalen, T. Bushmaker, D. H. Morris, M. G. Holbrook, A. Gamble, B. N. Williamson, A. Tamin, J. L. Harcourt, N. J. Thornburg, S. I. Gerber *et al.*, "Aerosol and surface stability of sars-cov-2 as compared with sars-cov-1," *New England Journal of Medicine*, vol. 382, no. 16, pp. 1564–1567, 2020.

[4] V. Venkatesh, "Impacts of covid-19: A research agenda to support people in their fight," *Int. Journal of Information Management*, p. 102197, 2020.

[5] TraceTogether App, Government of Singapore, https://www.tracetogether.gov.sg/.

[6] Aarogya Setu Mobile App, National Informatics Centre, India, https://www.mygov.in/aarogya-setu-app/.

[7] S. Meixner, "Phone scans, gps tracking and wristbands: How other countries do covid-19 contact tracing," https://www.abc.net.au/news/2020-04-28/coronaviruscovid19-contact-tracing-apps-around-the-world/12189438, 04/27/2020.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018, pp. 1–15.

[9] The Ordering Service, hyperledger fabric, Available at: https://hyperledger-fabric.readthedocs.io/en/release-2.2/.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf, 2008.

[11] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 297–315.

[12] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[13] D. Larimer, "Delegated proof-of-stake consensus," 2018.

[14] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv preprint arXiv:1906.11078*, 2019.

[15] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," 2018.

[16] H. Xu, L. Zhang, O. Onireti, Y. Fang, W. B. Buchanan, and M. A. Imran, "Beeptrace: Blockchain-enabled privacy-preserving contact tracing for covid-19 pandemic and beyond," *arXiv preprint arXiv:2005.10103*, 2020.

[17] J. Song, T. Gu, X. Feng, Y. Ge, and P. Mohapatra, "Blockchain meets covid-19: A framework for contact information sharing and risk notification system," *arXiv preprint arXiv:2007.10529*, 2020.

[18] D. Marbouh, T. Abbasi, F. Maasmi, I. Omar, M. Debe, K. Salah, R. Jayaraman, and S. Ellahham, "Blockchain for covid-19: Review, opportunites and a trusted tracking system," 2020.

[19] S. Haig, "Dutch govt to embrace blockchain in fight against pandemic," Cointelegraph, 2020.

[20] B. Magazine, "Blockchain and crypto firm vechain utilized to confirm authenticity of coronavirus kn95 masks," https://www.blockchainmagazine.net/blockchain-and-crypto-firm-vechain-utilized-to-confirm-authenticity-of-coronavirus-kn95-masks/, 2020.

[21] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.

[22] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy, "Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders," *Govt. Technology Agency-Singapore, Tech. Rep*, 2020.

[23] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, pp. 157 113–157 125, 2019.

[24] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.