Solutions for the Corda Security and Privacy Trade-off: Having Your Cake and Eating It

Tommy Koens, Scott King, Matthijs van den Bos,
Cees van Wijk, Aleksei Koren

ING
firstname.lastname@ing.com

Abstract. Corda is one of the major global distributed ledger technology (DLT) platforms. Currently, however, a trade-off has to be made in Corda between a particular security concern and a particular privacy concern. In this white paper we argue that this trade-off must be addressed as both concerns may have significant impact on Corda participants. We discuss potential solutions that address either this security concern or this privacy concern. We argue that current solutions that address the particular security concern are not preferred. These solutions either transfer the concern to another party or limit the functionalities offered by Corda. We conclude that either a hardware based approach or cryptography based approach are preferred solutions to the current security and privacy trade-off in Corda.

1 Introduction

Corda is a private permissioned distributed ledger and has attracted the attention of various industries, including energy, healthcare and capital markets [16]. The current Corda platform (version 4.1 [1]), however, requires participants to make a trade-off between security and privacy. All Corda network participants, at some point in time, are required to agree upon the new state of the ledger. This can be achieved via either validating notaries or non-validating notaries. Validating notaries consist of one or more nodes that reach consensus on the new state of the ledger. However, currently, the content of every transaction is revealed to a validating notary to be able to achieve consensus. Being able to observe the content of transactions may raise privacy concerns. Alternatively, non-validating notaries can also agree on the new state of a ledger without having to reveal to these notaries the entire transaction content. Although this increases both the number of transactions per second that can be processed as well improving privacy, it opens the possibility for a denial-of-state (DoSt) attack. In this attack a malicious actor is able to block the assets of a participant on the Corda network. Clearly, there currently is a trade-off to be made in Corda between security and privacy.

In this white paper we discuss this security and privacy trade-off. In section 2 we provide background information on Corda that contributes to our further discussion. We discuss the security and privacy trade-off that currently has to be made in Corda in Section 3. We argue that addressing the privacy concern is important as it could lead to operational and compliance risks for Corda participants. Furthermore, we argue that addressing the security concern is important as it is a known vulnerability in the current Corda design that can be exploited. Once exploited, Corda can no longer guarantee its primary purpose, which is to provide a "distributed ledger made up of mutually distrusting nodes ... allowing for a single global database that records the state of deals, obligations and other agreements between institutions and people" [2]. Additionally, the security concern could lead to a denial of services which are provided by Corda participants. Therefore, it is essential to address both concerns and the trade-off should not be dismissed. We examine solutions that address the security concern in Section 4. Obviously, we focus on solutions that address either concern without introducing the other concern, these are:

- Accepting the risk of the denial-of-state attack
- Off-ledger approach
- Introducing state signatures

We also examine solutions that address the privacy concern in Section 4, these are:

- Wet-signatures contracts
- Trusted Execution Environment (TEE)
- Zero-Knowledge Proofs (ZKP)

We argue in Section 4 that addressing the security concern is not a viable solution to the security and privacy trade-off. Addressing the security concern either shifts the security concern to another party or severely limits the functionalities offered by Corda. We provide our conclusions in Section 5. First, we conclude that the security and privacy trade-off in Corda must be addressed. Second, both the TEE as well as the ZKP solution seem most viable. Third, the SGX solution has a minor disadvantage (i.e. potential vendor lock-in) whereas the ZKP solution has a slight advantage (i.e. potential increase of transaction throughput). However, both solutions are viable and must be considered in addressing the current security and privacy trade-off in Corda.

2 Background

Corda is a blockchain-inspired open source distributed ledger platform [15]. Corda is a private and permissioned distributed ledger. Private means that it allows for a limited set of known participants to exchange assets. Furthermore, a limited set of known participants can reach consensus on the states (i.e. permissioned). A state is a fact registered on ledger. To change states, Corda participants can send transactions which propose new state changes. Such communication is based on a point-to-point protocol and there are no global broadcasts. This means that states are not widely shared [6].

A transaction consists of several components, as shown in table 1, including input states and output states. In a transaction, input states are consumed and transformed in a new set of outputs. For example, consuming an input state shows ownership of an asset by a participant, whereas an output state sets new ownership of that asset. To prevent an input state to be consumed more than once, the so-called double spend, notaries are also present on the Corda platform. A notary is an authority in the Corda platform with the aim to provide consensus.

2.1 Corda consensus

Consensus can be reached over state validity and state uniqueness [12].

State validity. A state is valid when a transaction has correct input and output states (i.e. they are valid according to the rules of the smart contract for this transaction), and the transaction has all required signatures which includes the signatures of the sender of the transaction and that of the notary cluster. A valid state allows parties involved in the transaction to be certain that a transaction is accepted by a contract that the transaction points to.

State uniqueness. Each state has only been approved once by a notary.

Table 1 shows which specific transaction components are revealed to each type of notary [13]: Note that the input states revealed to the non-validating notary is a composition of the transaction ID and the position of the state in the outputs. The kind of state nor its contents are revealed. Both type of notaries record the identity of the calling party, as prior to each transaction a notarization request [3] is send from the calling party to the notary. In this request the identity of a participant is stated. The identity consists of a public key and an X.500 Distinguished Name [13].

Table 1. Transaction components in Corda

Transaction components	**Validating notary**	**Non-validating notary**
Input states	Fully visible	References only
Output states	Fully visible	Hidden
Commands (with signer identities)	Fully visible	Hidden
Attachments	Fully visible	Hidden
Time window	Fully visible	Fully visible
Notary identity	Fully visible	Fully visible
Signatures	Fully visible	Hidden

3 Problem description

In Section 2 we described two ways in which a Corda network reaches consensus. The first approach to reach consensus is through validating notaries, the second approach is through non-validating notaries. In what follows we discuss how consensus is reached in both approaches, and we discuss the benefits and problems of each approach.

3.1 Validating notaries - a privacy concern

Notaries that validate transaction content before reaching consensus are called validating notaries. Validating notaries reach consensus after (a.o.) validation of (i.) the correctness of transaction inputs and outputs, and (ii.) the required signatures of all participants to the transaction Currently a validating notary can only validate a transaction when it is able to observe the full contents of the transaction, see Table 1, and must also be able to observe transaction dependencies, i.e. the entire transaction history up to the point of asset issuance. Being able to see the full transaction content could potentially be a privacy issue, as a transaction may contain sensitive (e.g. customer) data resulting in compliance risks, or market data can be derived (and acted upon) from the transaction history and volume (which could lead to operational risks).

3.2 Non-validating notaries - a security concern

In contrast to validating notaries, non-validating notaries can observe a limited amount of transaction information. As shown in Table 1 only the references ($_r$) of input states, the time window and notary identity are revealed to a non-validating notaries. A non-validating notary stores an ordered list of spent states (O). Once a transaction is send to a non-validating notary it verifies if the state O_r to which the transaction references has been spent. If the state has not been spent, the ordered list is updated. Also, any transaction that attempts to spend O_r from this point on is rejected. This implies that a non-validating notary can only determine if the previous state to which the input of a transaction points to is spent. It is important to note that non-validating notaries do not verify signatures.

Although revealing limited information addresses the privacy concern that exists with validating notaries, it does currently allow for a denial of state attack. In what follows we explain how this attack is executed and under which assumptions.

Denial-of-state attack. First we list the assumptions made for a successful denial-of-state attack, which are:

- A malicious actor exists who wishes to perform a denial-of-state attack
- A Cordapp that uses a non-validating notary exist.
- The malicious actor must have access to the Corda network. A malicious actor can be either an insider or outsider. An insider is any participant that was granted access to the Corda network which could be, for example, a disgruntled employee. An outsider is anyone who was not granted access to the Corda network but managed to gain access through compromising a Corda node.

- The malicious actor knows the reference to a particular state S_i.
- The malicious actor is able to create a transaction T_i that consumes state S_i and sends this to the non-validating notary.
- The non-validating notary considers the transaction T_i to be valid and consumes state S_i.

Note that two practical and crucial assumptions are present. First, the malicious actor must have access to the Corda network. Second, the malicious actor must know the details of the state S_i.

Under these assumptions a malicious actor can perform a denial-of-state attack.

The malicious actor can perform the attack by creating and signing a transaction with its private key. Non-validating notaries do not verify signatures, but do verify if the state can be consumed. When the state is consumed the pointer to the actual transaction data is no longer valid (as it is consumed). From this we can conclude that a denial-of-state attack results in:

- A particular state can no longer be spent
- A state can no longer be spent by its owner
- The current owner is not aware that the state cannot be spent, up until the moment when the current owner attempts to spend the state
- A denial-of-state attack can thus remain undetected for a long time, until an attempt is made to spend the state
- The previous state can not be spent by the malicious actor.

3.3 Summary

Under the assumption that both security and privacy has to be achieved when reaching consensus in a Corda network, currently, there clearly is a trade-off to be made between security and privacy. Also, as argued in Section 1, both the security concern as well as the privacy concern must be addressed because of their potential consequences to both the Corda platform as well as its participants.

4 Potential solutions and discussion

In this section we discuss potential solutions to these concerns. In Section 3 we discussed both the security concern and privacy concern that may arise when a Corda network reaches consensus. We argued that currently a trade-off has to be made between securty and privacy. In principle there are two approaches to address the trade-off. First, we can address the security concerns without re-introducing the privacy concerns. Second, we can address the privacy concerns without re-introducing the denial-of-state attack. In what follows we discuss potential approaches for both concerns.

Addressing the security concern. In this section we address the security concern that appears when privacy is present in a Corda network. The current main security concern in Corda is the denial-of-state attack, as discussed in Section 3. There exist both internal as well as external threat actors that may be the cause of a denial-of-state attack. For example, a disgruntled employee could have access to the Corda network and initiate such an attack. Another example is where a trusted party on the Corda network is breached by a malicious actor. Therefore, the possibility of a denial-of-state attack is a concern that must be included in any Corda threat model. Furthermore, we can assume that a successful attack may have significant consequences in particular use case scenarios, as discussed in Section 1.

Accepting the risk of a denial-of-state attack. Accepting the risk of a denial-of-state attack should be assessed on a case-by-case basis. However, risk acceptance leaves any Corda network that uses non-validating notaries vulnerable to a known attack which may lead to significant business disruption, depending on the use case. Addressing the issue seems called for, as it solves the vulnerability for *any* Corda network which uses non-validating notaries.

Off-ledger approach. An off-ledger approach is to, after a denial-of-state attack occurred, to roll-back to the state of the ledger *before* the attack occurred. This is a responsive measure to a denial-of-state attack and requires the otherwise immutable history of the ledger to be adjusted. This clearly goes against the concept of immutability, as well as a Corda concept which is providing ownership of state. Namely, ownership of state is no longer guaranteed when another party is able to adjust a state of which it is not the owner. In fact, we argue that handling denial-of-state attacks off ledger only shifts the problem to another party (or set of parties). Namely, the party re-writing the history of the ledger to its correct state after a denial-of-state attack occurs is able to alter states.

A variant of this approach would be that a majority of network participants are able to vote to reassign ownership of a state. This approach transfers the security issue to another set of parties, similar to the roll-back scenario described above.

Clearly, responsive measures to a DoSt-attack are not desired. Therefore, preventive measures are called for to address the DoSt-attack.

Introducing state signatures. A preventive approach would be to assign signatures to a state, which proves state ownership. This requires a functional change on the non-validating notaries as they have to start verifying signatures. State ownership then can only be changed when the current owner is in possession of the private key that is related to the signature assigned to a state. Although this addresses the denial-of-state attack, it severely limits the functionalities Corda offers and does not fit into its current architecture. From a functionality perspective, for example, a smart contract may state that change of state ownership requires a single signature before a particular date. The same contract may hold a rule that states that after the particular date, two signatures are required. However, a notary would only verify the state ownership based on a single signature, as only a single signature is assigned to the current state.

From an architectural perspective, for example, notaries currently verify some transaction components. It would be possible to shift the smart contract verification towards a notary, but this would require a shift in architectural design, as the smart contract verification currently is done by the network nodes. A major downside here is that the increased load for notaries which may impact scalability.

Following this discussion, we conclude that when we wish to achieve both privacy and security in Corda, addressing the security concerns when privacy is present may not be the best approach.

Addressing the privacy concern. In this section we discuss how privacy concerns can be addressed under the assumption that security is present.

Wet-signature contracts. The main idea is to use validating-notaries in a Corda network which are able to observe the content of all transactions. To address this concern wet-signature contracts can be drafted between participants on the Corda network. For example, if information leaks from one of the notaries, its owner can be sued and fined. However, there are three issues here:

- First, it may be hard to determine which notary leaked the information as in a notary cluster typically the information is shared among notaries. Additional (operational and technical) measures should be taken to monitor the notaries for information leakage.
- Second, compliance rules require that some confidential information (e.g. customer data) is protected beyond wet-signature contracts only. Following our assumption (i.e. both security and privacy are needed in a Corda notary) at the start of this section, wet-signature contracts are insufficient to address the privacy concern.
- Finally, it would be possible to argue to use a centralized solution instead of a distributed ledger. Any concerns then will be addressed by wet-signature contracts, making a distributed ledger obsolete. This is likely uncalled for, as the parties involved in the Corda network have a common aim, namely, improving their business by means of Corda.

Trusted Execution Environment (TEE) TEE are trusted hardware that allow for private processing of data. A specific example of TEE is SGX, which is a promising solution by Intel that addresses privacy concerns. However, SGX, or any hardware based privacy solution, is subject to a few issues of which we name three:

- Third party lock-in. As TEE are provided by specific hardware vendors, it would be possible for such a vendor to claim a foot-hold and thus create a vendor lock-in.
- SGX is hard to implement in existing corporate infrastructure. Although new infrastructure can be standardized by incorporating SGX (and risk third party lock-in), replacing existing hardware is typically a long process in corporate organizations.

- Arms-race. There is an ongoing arms-race where the security design of SGX is being tested. There are numerous examples of attacks, for example [10], [8], and [9], and corresponding vendor patches [7].

Zero Knowledge Proof (ZKP). ZKP is a cryptographic method where one party can prove to another party to be in possession of some knowledge, without revealing that knowledge. Originally proposed in 1989 by Goldwasser, Micali, and Rackoff [4], this field of cryptography has received a lot of academic attention and is battle tested in practise, for example ZCash [5], a privacy-friendly cryptocurrency, employs a variant of ZKP. ZKP can also be used to address privacy concerns in Corda notaries. Namely, it would be possible for the owner of a state to prove ownership to a notary, and it would allow for proving that the state has been transferred. Additionally, surprisingly, ZKP transactions that are sent to notaries take less time to verify than regular transactions (of which the entire content is visible) [11]. However, cryptographic solutions also introduce some issues:

- Quantum computing. Most of current cryptographic principles are no longer to be secure with quantum computing. Developing a ZKP solution for Corda should take this into account too.
- Battle testing specific implementations. A ZKP solution for Corda notaries should be properly verified (i.e. formal verification), tested (i.e. implement in a test-environment), and reviewed (e.g. academic peer review). Such a process is not trivial and may take time to complete which may contrast a corporate environment aiming for swift production-ready implementations.

Transaction tear-offs. This concept has been proposed in Corda to increase privacy, by showing only the minimum amount of information and 'tearing-off' the information that should be kept confidential from the transaction [14]. As such, transaction tear-offs could be applied in combination with a validating notary. We discuss two scenarios:

- Scenario 1. Apply transaction tear-off so that only the transaction components are shown similar to that of a non-validating notary (see Table 1). Clearly, this reduces a validating notary to a non-validating notary and introduces the DoSt attack, which we denoted as a security concern as discussed in Section 3. At its core, a DoSt attack is possible because the non-validating notary does not verify the transaction signatures, which brings us to the second scenario.
- Scenario 2. Apply transaction tear-off so that only the transaction components are shown similar to that of a non-validating notary *and* include the transaction signatures (see Table 1). This allows the validating notary to validate the identity of a participant, which prevents the DoSt attack. In this scenario, however, a validating notary would be able to create an audit trail of signatures from a particular participant, which introduces the privacy concern.

Therefore, we do not consider transaction tear-offs to be a viable solution as it introduces either a privacy concern or a security concern, as discussed in Section 3.

5 Conclusion

Although each use case built on Corda may have different requirements when it comes to privacy and security, in this work we argued that the trade-off between security and privacy in Corda notaries should be addressed. When privacy is chosen, notaries are not able to see the content of transactions. However, this introduces a security issue, namely the denial-of-state (DoSt) attack. We argued that the DoSt-attack can not simply be dismissed and must be considered a security threat, as discussed in Section 4. When instead security is chosen, this attack can no longer occur. However, notaries are able to see the full content of transactions. We argued that the lack of privacy within a Corda network, when validating notaries are employed, also may raise concerns in Section 4. Therefore, both security *and* privacy should be addressed.

As Corda currently either addresses the security concern *or* the privacy concern, we can approach this trade-off in two ways. First, we can address the security concern with the notion that no privacy concern is present. Second, we can address the privacy concern under the notion that the security concern is not present. We investigate what approaches could address this trade-off between security and privacy in Corda.

Addressing the security concern may not be the best approach as it either shifts the concern to another party or limits the functionalities offered by Corda, as discussed in Section 4. Addressing the privacy concern may be a better approach and several approaches exist, as discussed in Section 4. We discussed wet-signature contracts, Trusted Execution Environments (TEE), and Zero-Knowledge proofs (ZKP). We conclude that both the TEE as well as the ZKP approach provide a solid solution to the security and privacy trade-off in Corda that should be addressed.

References

1. Katelyn Baker. Corda 4.1. 2019. `https://github.com/corda/corda/releases`.
2. Richard Brown. The Corda Platform: An Introduction. 2018. `https://www.corda.net/content/corda-platform-whitepaper.pdf`.
3. Corda. Notarisationrequest. 2018. `https://docs.corda.net/api/kotlin/corda/net.corda.core.flows/-notarisation-request/index.html`.
4. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.
5. Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification. *Tech. rep. 2016–1.10. Zerocoin Electric Coin Company, Tech. Rep.*, 2016.
6. hyperledger. Corda Documentation. 2018. `https://buildmedia.readthedocs.org/media/pdf/corda/latest/corda.pdf`.

7. Intel. Intel SGX Patchwork. https://patchwork.kernel.org/project/intel-sgx/list/.
8. Yeongjin Jang, Jaehyuk Lee, Sangho Lee, and Taesoo Kim. Sgx-bomb: Locking down the processor via rowhammer attack. In *Proceedings of the 2nd Workshop on System Software for Trusted Execution*, page 5. ACM, 2017.
9. Tommy Koens. Consensus by trusted hardware. 2018. https://www.linkedin.com/pulse/consensus-trusted-hardware-tommy-koens/.
10. Esmaeil Mohammadian Koruyeh, Khaled N Khasawneh, Chengyu Song, and Nael Abu-Ghazaleh. Spectre returns! speculation attacks using the return stack buffer. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.
11. Eduardo Morais, Tommy Koens, Cees van Wijk, and Aleksei Koren. A survey on zero knowledge range proofs and applications. *SN Applied Sciences*, 1(8):946, 2019.
12. R3. Consensus and notaries. 2016. https://docs.corda.net/releases/release-M9.2/key-concepts-consensus-notaries.html.
13. R3. Notaries. 2018. https://docs.corda.net/key-concepts-notaries.html.
14. R3. Transaction tear-offs. 2018. https://docs.corda.net/key-concepts-tearoffs.html.
15. R3. Welcome to Corda! 2018. https://docs.corda.net/releases/release-V3.0/.
16. R3. Customers. industry success stories. 2019. https://www.r3.com/customers/.