2PC is for P2P: A Decentralized P2P Marketplace for Trading Digital Goods

Sachin Kumar, P2P Labs Inc.

Special thank you to Yuki Yuminaga and Fenbushi Capital for sponsoring the research through grant support. This paper was presented at ZK Summit 11, Athens in April 2024. You can view the recording here - ZK11: 2PC is for P2P: A TLSN-Based On/Off-Ramping Protocol

Abstract

In this paper, I propose a novel protocol to enable trustless peer-to-peer (P2P) trading of digital goods. The protocol leverages TLSNotary ("TLSN") for verifying off-chain transfers of digital goods and an on-chain escrow-based smart contract protocol to manage payments. Additionally, it introduces a network of notaries, bootstrapped by an Eigenlayer Actively Validated Services ("AVS"), that guarantees decentralization and liveness to support the trade to its completion. The solution employs Optimistic Notarization with a single notary within this network to ensure quick and effective notarization. Similar to optimistic rollups, the protocol allows the counterparty in the transfer to challenge notarizations in cases of suspected collusion, thus initiating the arbitration process facilitated by the network. If the counterparty successfully proves malicious behavior, it leads to the economic slashing of the offending notary. This mechanism deters notaries from acting maliciously, effectively solving the collusion problem in TLSN.

1 Introduction

1.1 ZKP2P

ZKP2P is a trust-minimized P2P fiat on and offramp interoperable with all popular web2 payment rails (e.g. Venmo, UPI, Revolut). Our initial protocol is powered by zero knowledge ("ZK") proofs of DKIM signatures in payment confirmation emails, which allows for permissionless integration with any web2 payment network. ZKP2P allows users to unlock escrowed assets on-chain in a trustless manner after a successful off-chain payment and proof generation.

Currently, it is challenging to find suitable and data-rich payment confirmation emails that fit our protocol's need for generating ZK proofs. Additionally, the non-standard and frequently changing email templates can necessitate regular updates to ZK circuits.

Therefore, we are interested in new areas of attested data that can better serve our use case at ZKP2P. In particular, APIs are more data-rich, stable, and are almost always backward compatible. However, unlike emails that are signed by DKIM, APIs are often not signed, which means we require an alternative approach to prove the provenance of API responses.

1.2 TLSNotary

The TLSNotary protocol [1], developed by the Privacy and Scaling Explorations (”PSE”) research lab of the Ethereum Foundation, enables data portability, allowing a user, the Prover, to share it with another party, the Verifier, as desired. This is achieved through applied cryptography techniques and TLSN, which is widely utilized on the web today.

The vanilla TLSN protocol consists of three actors: the Prover, Verifier, and Server. It operates as follows:

1. The Prover requests data from a Server over TLS while cooperating with the Verifier in secure multi-party computation (”MPC”).
2. The Prover then selectively discloses the data to the Verifier.
3. The Verifier verifies the data.

The TLSN protocol is essentially divided into two sub-protocols. The first is the MPC-TLS protocol, which secures a private and authenticated commitment to the data from the Verifier. The second is the selective disclosure protocol, designed to verify data properties while maintaining privacy. Together, these protocols fulfill TLSN’s primary objective of enabling data portability while preserving privacy of any sensitive information.

The MPC-TLS protocol ensures that nothing about the plaintext of the TLS session or the Server’s identity is disclosed to the Verifier. This allows for the outsourcing of the MPC-TLS session to a general-purpose verifier, referred to by the TLSN team as the Notary.

The Notary executes the MPC-TLS protocol alongside the Prover and blindly signs a commitment to both the data and the server’s identity. This signature renders the data reusable and portable. Subsequently, the Prover can present this signed data to an application-specific verifier, who then authenticates the data.

2 Problem

The TLSNotary protocol, while facilitating data portability and secure sharing between a Prover and a Verifier, does not address the oracle problem. The core issue lies in the application-specific verifier’s reliance on the Notary’s integrity. There is an inherent risk of collusion between the Notary and the Prover, where a malicious Notary could authenticate forged sessions—sessions not genuinely conducted between the client and the server.

To solve the TLSN collusion problem for the general case, one could propose leveraging decentralized trust, which involves spreading the notarization process across a network of notaries, underpinned by an honest majority trust assumption. However, implementing the MPC-TLS protocol in a decentralized context presents significant challenges. Given that the protocol relies on two-party computation ("2PC"), which is bandwidth-intensive, extending this process to a large, decentralized network of nodes would result in substantial inefficiencies and elevated costs.

3 Solution

To solve the TLSNotary collusion problem specifically for the peer-to-peer (P2P) trading case, we initially consider a solution where the counterparty in the trade acts as the Notary. This approach is particularly suitable given our unique scenario, where the transaction involves two parties positioned at opposite ends of the trade. By exploring this initial solution, we identify its inherent issues and fully understand its limitations and potential challenges. Following this analysis, we then develop and propose a refined solution that addresses these identified problems, ensuring a more secure and reliable framework for P2P trading.

To start, let's delineate the roles within the TLSN protocol as they apply to our P2P marketplace protocol. In the TLSN context, the Prover possesses off-chain data and is capable of verifying its origin and characteristics. Translated to our P2P marketplace, the Prover equates to the digital goods seller, responsible for initiating the off-chain transfer of goods. Meanwhile, the digital goods buyer locks their funds on-chain within a smart contract as a form of bid. The seller has the option to select their preferred bid and proceed with the sale. This transaction process entails the seller transferring the off-chain digital goods to the buyer and generating a proof of transfer. This proof then facilitates the release of the funds held in escrow.

In the described scenario, designating the buyer as the Notary introduces a safeguard against collusion between the buyer (Notary) and the seller (Prover). A rational buyer acting as the Notary is unlikely to collude with the seller to create a false proof, as this would jeopardize their own assets, which are already locked in the on-chain escrow contract. Any such collusion could be identified as wash trading, indicating that the inherent interests of a buyer serving as a Notary naturally mitigate the risk of fraudulent activities.

However, implementing this solution as a protocol introduces several practical challenges:

1. **Operational Complexity:** Buyers have an incentive to keep their notary services running primarily to benefit from the transaction spread. However, the expectation for all buyers to operate a notary could lead to a reduction in the protocol's liquidity and transaction volume, highlighting a significant concern from a product perspective. The focus on developing

trust-minimized systems often overshadows the need to accommodate the practical needs and preferences of users, which will be detrimental to the marketplace.

2. **Liveness, Arbitration, and Slashing:** The protocol faces a significant liveness issue if the buyer's notary goes offline or censors notarization, after receiving off-chain digital goods, preventing the seller from proving the transfer and accessing the escrowed funds. This is ideally something we would like to avoid completely but let's accept it for now. In such liveness failure cases, a fallback option—either a centralized notary or a decentralized network of notaries—could perform the notarization, acting as an arbitration mechanism. Resorting to a centralized notary for arbitration doesn't align with our objective of minimizing trust, as it merely shifts the trust issue to a centralized entity. Arbitration by a decentralized network of notaries, under an honest majority assumption, is a more acceptable solution provided arbitration is infrequent and the financial burden falls on the malicious party.

However, the requirement for inexperienced users to operate notaries could lead to more frequent liveness failures and, consequently, arbitration, which is undesirable. Moreover, there is no way to prove a malicious notary's liveness, hindering the ability to penalize the malicious behaviour.

Thus, the above-mentioned solution, beyond the operational complexity, grapples with a liveness failure issue, for which the fallback arbitration process lacks a definitive mechanism to safeguard honest participants and penalize malicious conduct.

4 Proposed Solution

To address the issues of liveness, operational complexity, and punishing malicious behavior identified in the initial construction, we propose delegating the notary's responsibilities to an operator within a network. Notarization is performed by a single designated operator which is cheap, efficient, and feasible using TLSN today. We reuse the above insight of leveraging our unique P2P trading case, but in this solution, the buyer's requirements are relaxed and they just have to verify the accuracy of the notarization and ensure the notary operator has not colluded with the seller (Prover). The proposed solution is similar to optimistic rollups, where we optimistically assume that the single chosen notary did not collude with the prover. Moreover, this optimistic notarization approach is strengthened by the presence of a counterparty for each transaction, who is directly incentivized to initiate the arbitration process to avert their own financial loss.

4.1 Notary Operators

The responsibility of running the notary is delegated to operators in this solution. This ensures everyone can participate in the P2P marketplace. The network of notaries could be bootstrapped using a solution such as an Eigenlayer AVS. The AVS helps bootstrap nodes from the large validator set of Ethereum. These validators restake their ETH in Eigenlayer and opt-in to notarize user sessions. If a validator who is restaked in EigenLayer is proven to have behaved adversarially while participating in an AVS, then the staker's ETH will be subject to slashing according to the on-chain slashing contract of the AVS. Notary operators participating in regular notarizations can be compensated by the fees earned by the marketplace protocol. Operators participating in an arbitration process can be compensated from the slashed amount.

4.2 Liveness and Censorship Resistance

The client can select the notary for the session randomly from the set of active operators. However, the MPC-TLS protocol run between the Prover and Notary is mostly input/output ("IO") bound and its reliability and performance require a low latency connection between the two. Thus, until TLSNotary implementation's is improved and the bandwidth requirements are reduced, the client could select the notary closest to the prover and with good upload and download speed. This deterministic selection could be abused by the prover to a certain degree. In case the chosen notary is not live or purposefully refuses to notarize the session, the prover can always request another notary in the network. Hence, they are never stuck after completing the off-chain transaction. Censorship-resistance and liveness guarantees can be achieved if the network is sufficiently decentralized and a large number of validators have opted-in to the EigenLayer AVS [2].

4.3 Arbitration and Slashing

If the buyer detects any discrepancy and thinks the seller (Prover) and notary operator have colluded, they can initiate an arbitration process and challenge the previous notarization and corresponding on-chain escrow unlock. The arbitration process requires the challenger, in this case, the buyer, to run the MPC-TLS protocol with the majority of the nodes in the network. The challenger would notarize a specific API response or webpage that proves that they didn't receive the digital goods that their counterparty claims to have transferred to them. Once the challenger receives notarizations from a majority of the network they can aggregate all of them into a single proof that can then be submitted on-chain to slash the operator which performed the fraudulent notarization. Given that that escrow had already been optimistically unlocked before, the proceeds from slashing will be used to compensate the challenger (buyer). The challenge period can be 7 days, similar to optimistic rollups.

This slashing mechanism puts a cap on the max net value of assets that can be traded safely through this protocol per transaction. If we use EigenLayer to bootstrap the network of notaries, each notary operator has a maximum current stake of 32 ETH. Hence we can't provide a complete economic guarantee for a P2P transaction where the value of goods traded is above 32 ETH. Also, the requirement that the challenger needs to provide a certain API response or webpage that proves they didn't receive the digital goods prevents certain types of digital goods from being supported by this protocol. We have observed that most financial apps provide a list of transactions to the receiver that can't be modified by the user, thus on/off-ramping can be supported on this marketplace.

The arbitration process involving a majority of the nodes in the network will be costly and inefficient. However, malicious behavior poses significant risks and severe economic penalties, with a high certainty of being caught by the counterparty, which deters them from colluding and acting maliciously in the first place. This behavior has precedence, for example, since its mainnent launch in August 2021, Arbitrum has not received any fraud-proof submissions [3]. Thus, despite the potential for high costs and inefficiencies, the arbitration process can be assumed to be rarely invoked.

4.4 Privacy

The seller and the third-party notary run the MPC-TLS verifier protocol which guarantees complete data privacy to the seller. The notary blindly signs a commitment to the data and the server's identity. The prover can perform the selective disclosure and server identity verification inside a zk-snark circuit run on the client side, and post the generated proof on-chain to unlock the escrowed funds on the smart contract.

Thus by introducing a network of notary operators bootstrapped by EigenLayer AVS with staked capital, we address the three primary concerns of the initially proposed solution. The proposed solution reduces operational complexity by delegating the notary operations to dedicated operators with proven uptime records, thus mitigating the risk of liveness failures and censorship resistance. Additionally, it ensures enforceable economic penalties in case of collusion through guaranteed slashing mechanisms. The use of optimistic notarization aligns with current technological capabilities, offering a user-friendly experience for marketplace participants, thereby making it feasible to implement this solution promptly with existing tools.

5 Future Work

The proposed solution aims to reduce the likelihood of initiating the arbitration process. However, the possibility of arbitration remains, and a reliable and efficient method for conducting it needs to be clearly defined and specified. The challenge is exacerbated by the globally distributed notaries, given that the MPC-TLS protocol is heavily input/output bound.

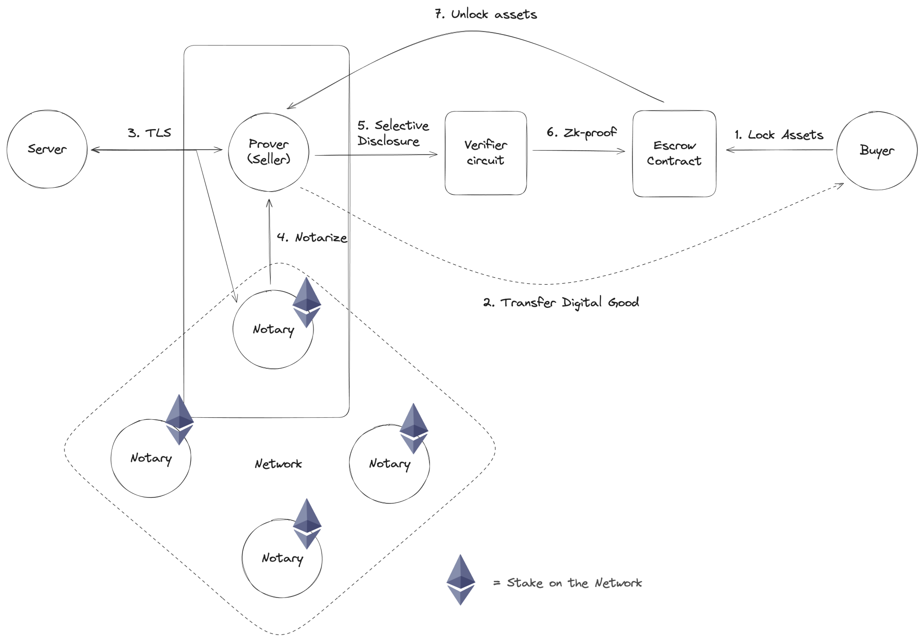

Figure 1: Notary Network

Currently, the specifics of the slashing conditions are not fully established. In scenarios where the protocol facilitates numerous trades and a single notary engages in multiple fraudulent activities before being challenged, the recovered funds from slashing may not suffice to compensate all affected parties. To address this, consideration is being given to limiting the maximum value of digital goods that can be traded in each transaction.

References

1. TLSNotary Documentation (2024)

2. Eigenlayer Documentation (2024)

3. Cointelegraph: Arbitrum Fraud Proofs (2023)