IEEE ICBC'24 1570978833

NiFT: A Non-interactive Fair Trading Protocol

***Abstract*—The volume of online data marketplace is growing rapidly with the continuously increasing value of data. Ensuring fairness between sellers and buyers is essential in online data marketplaces as they do not trust each other due to the nature of the Internet. With the emergence of blockchain technology, it is possible to guarantee online trading fairness without leveraging a single trusted authority. In existing solutions, data are verified first and then exchanged with cryptocurrencies atomically through an HTLC-based smart contract. Unfortunately, direct secure channels between users (buyers and sellers) are necessary for existing solutions, which are not widely available on the Internet. Relaxing the assumption of such secure channels increases the practicality since users do not need to set up such channels, but it is challenging to do so because some consensus must be secretly reached between users to deploy smart contracts. We address this issue by proposing a novel non-interactive fair trading protocol called NiFT that eliminates direct user communications. We analyzed our protocol security and proved that, except for sellers, other users cannot access the complete dataset unless they can solve the discrete logarithm problem in polynomial time. Evaluation results indicate that our protocol reduces communication costs for both seller and buyer, with some buyer-side overhead compared to existing solutions.**

***Index Terms*—Data trading; fair exchange; atomic exchange; non-interactive protocol**

I. Introduction

In the era of big data, both industry and academia can benefit from using data, either to improve their quality of service or to conduct better solutions to research problems. Data are considered a tradable asset as their value continuously grows, and data trading is becoming more and more prevalent. According to a recent report [1], the global data marketplace volume was more than $960 million in 2022, and the compound annual growth rate (CAGR) is estimated to be 25% from 2023 to 2030. In recent years, many laws and policies (e.g., GDPR, CCPA, CPRA, etc.) have also been proposed for better regulation and supervision, further proving the rapid development speed of the market.

Ensuring digital trading fairness, also known as the fair exchange problem, is vital as users do not trust each other due to the nature of the Internet. Two untrusted parties, a seller and a buyer, are usually considered in a fair exchange problem. An exchange between the seller and the buyer is considered fair if both parties receive what they want or nothing happens. Previous work [34] has proved that a trusted third party (TTP) is necessary to guarantee trading fairness. Traditional systems usually assume a single TTP responsible for executing transactions and/or resolving disputes but have a potential single-point-of-failure risk.

In recent years, the emergence of blockchain technology [33] has made it possible to let untrusted parties reach an agreement on some results without any single TTP. Many proposed works [15], [20]–[22], [28] use blockchain to ensure online trading fairness. These works usually assume the data are encrypted. Instead of relying on the previously assumed single TTP in the middle, both parties will exchange decryption keys and cryptocurrencies and/or solve disputes through smart contracts, which are publicly executed on the blockchain. The buyer usually deploys a Hash Time-Locked Contracts (HTLC)-based smart contract to lock some cryptocurrencies into the smart contract as a deposit. To claim the deposit, the seller has to provide a valid value with which the buyer can recover the decryption keys to the encrypted data.

As all the blockchain nodes will execute the smart contract, the input value provided by the seller is publicly available. Therefore, if the seller provides the key directly to the smart contract, the encrypted data must be transferred secretly [21], [22], [28]. Otherwise, other users can also decrypt and access the data. Alternatively, sellers and buyers may secretly share some values so that the buyer can generate a smart contract based on them [15], [19], [20] while publishing the encrypted data. Then, the seller provides the input such that only the buyer can recover the decryption keys. Either solution can ensure trading fairness, but a direct, secure channel between the seller and the buyer must be constructed before trading.

Although some form of secret sharing between sellers and buyers is necessary, building a pair-wise channel may cause extra processes/infrastructures, influencing the protocol's decentralization. Further, extra user effort/time is required for such channels (e.g., TLS handshake [3]). It is said that Amazon will lose 1% of its sales per 100ms latency [38]. Users tend to switch to another website if the current one is slower than its competitors by more than 250ms [32], and hence there are many works aiming at latency reduction [7], [9], [29], [37]. Besides, the communication in such channels is private, and thus, dispute resolution becomes significantly challenging as well since the communication is unobservable. Therefore, ensuring two parties can fairly and securely exchange payments and data access (e.g., the decryption keys or the value for recovering the keys) is challenging without having pair-wise secure communication channels between sellers and buyers.

In this paper, we present NiFT, a Non-interactive Fair Trading protocol. We remove the previous work's assumptions of pair-wise secure communication channels between sellers and buyers, thereby avoiding any direct communication between them. This is achieved by incorporating the previous secret-sharing process between the two parties into the smart contracts with cryptographic protocols. Due to the elimination of secure channels, a new technical challenge appears. Namely, some unauthorized users may be able to claim buyers' de-

posits. We present a novel solution to address the issue, and our solution causes no on-chain overhead. Our contributions are summarized as follows:

- We designed a non-interactive atomic exchange protocol based on HTLC. The seller and the buyer can fairly exchange data and payments without pair-wise secure communication channels, reducing their cost. The exchange process occurs on-chain only. Therefore, potential off-chain communication disputes are eliminated.
- We proposed a self-maintained storage service, and by following the protocol, sellers can prevent unauthorized users from claiming buyers' deposits.
- We proved the security of our non-interactive atomic exchange phase by contradiction/reduction, and the results indicate that our protocol is secure unless there is a polynomial time algorithm that can solve the discrete logarithm problem.
- We implemented our protocol and released anonymized source code [4] of our smart contracts for reproducibility. Simulation results show that NiFT reduces communication costs for both seller and buyer, with some buyer-side extra gas cost compared to existing solutions.

The rest of the paper is organized as follows: Section II introduces related works; Section III gives the preliminaries of the paper; Section IV describes the detailed design of NiFT, security analysis and discussion; Section V shows the evaluation results and Section VI summarizes the paper.

II. Related Works

Traditional approaches to fair exchange protocols have generally relied on a single trusted third party (TTP) [5], [6], making them vulnerable to single-point-of-failure issues. Some protocols operate on a bit-by-bit basis [8], [24], which is computationally inefficient. Given the inherent limitations of completely eliminating TTPs [34], recent advancements have focused on replacing a singular TTP with blockchain-based smart contracts. Fair data transfer protocols can be utilized for many applications, including financial fairness through auction methods [14], resistance to double-spend attacks via punishment transactions [25], and data leakage prevention during reselling [41]. These protocols often employ the concept of atomic swaps [27], originally inspired by Hash Time-Locked Contracts (HTLC) in the lightning network [36]. Atomic swaps are commonly used in cross-chain digital asset exchanges [11], [12], [18], [30]. In FairSwap [21], a specialized smart contract called *Judge* is invoked when the receiver disputes the received data. The *Judge* contract evaluates the complaint and, if valid, refunds the receiver. OptiSwap [22] extends this by introducing a challenge-response mechanism for dispute resolution. These works optimistically assume the exchange can be performed successfully (also named the optimistic mode) and focused on dispute resolution by using the multi-round interactive protocol on a blockchain between the buyer and seller, which could be non-practical in a data trading system.

In data trading scenarios, one common solution is to use certain mechanisms to let the buyer confirm the data (without revealing it) first and then initiate the exchange for the best efficiency. For example, Delgado-Segura et al. [20] offer a scheme that employs Bitcoin transactions for data exchange. In this approach, the buyer and seller negotiate the data and its price. The seller then encrypts and sends data slices to the buyer for verification. Upon successful verification, they exchange decryption keys and payment through a private key-locked transaction on Bitcoin [19]. Notably, the buyer can request the revelation of random encrypted slices for verification before proceeding with the transaction. Similarly, Zhao et al. [43] use Double-Authenticating-Preventing Signatures [35] (DAPS) for key exchange and allow data consumers to challenge portions of the encrypted data for verification.

III. Preliminaries

A. System/Security Models and Assumptions

For each exchange, we consider two mutual untrusted participants: A seller/sender $\mathcal{S}$ is an entity that will provide access to his/her data for those who have paid him/her. A buyer/receiver $\mathcal{B}$ is an entity that will pay for access to some desired data provided by a seller.

Note that in this paper, both payment execution and data transfer are considered to happen publicly online. Therefore, the data should be encrypted during transmission to prevent other users (who do not make payments) from accessing it, and the payments will be exchanged with the decryption keys. We then give the definition of fairness as follows:

Definition 1 (Fairness)**.** *We say an exchange between $\mathcal{S}$ and $\mathcal{B}$ is fair if: 1) $\mathcal{S}$ cannot get paid unless s/he provides the correct decryption keys to $\mathcal{B}$; 2) $\mathcal{B}$ cannot access the content of data unless s/he makes sufficient payment to $\mathcal{S}$. In other words, either 1) both parties get what they want (the exchange successfully happens), or 2) neither party releases/loses anything (the exchange fails), is considered fair in NiFT.*

Definition 1 follows the idea of *strong fairness* given in the previous work [31], where either both parties receive what they want, or the protocol is terminated with nothing exchanged.

For each exchange, if there is a dispute, either $\mathcal{S}$ or $\mathcal{B}$ will be considered malicious. Note that in case both $\mathcal{S}$ and $\mathcal{B}$ are malicious, where neither $\mathcal{S}$ will provide the decryption keys nor $\mathcal{B}$ will make payments, the exchange will not happen, which means the fairness is still ensured. Therefore, we only discuss how to ensure fairness if one of the two parties is malicious. As aforementioned, we assume that the data are transferred with encryption during each exchange, and at most one of the two entities will be malicious if there is a dispute between them. We also assume that the users can use existing approaches to validate whether the provided data are correct or not [15], [20], [21], [28]. Since the data verification is not mainly focused in NiFT, we choose some existing methods as building blocks and explain them in detail in Section III-C.

Besides, blockchain is introduced as a distributed third party to replace the single trusted party. Blockchain nodes/peers are responsible for transaction/smart contract verification and/or

execution so that an execution environment for our fair exchange protocol is guaranteed. Peers are considered semi-honest, meaning they will not deviate from the protocol but may try to infer some secrets during the execution (e.g., access the data). Other kinds of peers' misbehaviors, for instance, actively colluding with buyers to double-spend payments, are orthogonal to this work. This is because such misbehaviors are under the discussion of the underlying blockchain's security, e.g., consensus mechanism.

Therefore, with the above assumptions, we consider three types of adversaries in NiFT: 1) a malicious $\mathcal{S}$ may try to receive payments without providing the correct decryption keys; 2) a malicious $\mathcal{B}$ may try to get access to the decryption keys (or directly to the data) without paying anything; and 3) semi-honest blockchain peers will correctly execute the protocol but try to infer the decryption keys based on the publicly-available information (e.g., the smart contract) during the trading process. Note that malicious behaviors of $\mathcal{S}/\mathcal{B}$ during the data verification process are not considered in NiFT, as they are orthogonal to our scenario, and existing approaches [15], [20], [21], [28] can be further leveraged. Besides, this paper does not discuss cryptanalytic attacks initiated by more powerful adversaries, such as quantum computers.

B. Cryptographic Primitives and Building Blocks

(DDH) problem and Discrete logarithm problem. A DDH (Decisional Diffie-Hellman) problem is to, in a finite group $\mathbb{G}$ with generator g, given a triple (g^a, g^b, g^c), decide whether g^c is g^{ab} or a random element in $\mathbb{G}$ where a, b, c are all integers. A discrete logarithm problem is that in a finite group $\mathbb{G}$ with generator g, given a group element $y \in \mathbb{G}$, calculate an x such that $g^x = y$. Many finite groups exist where the DDH problem is known to be intractable [10]. In fact, if the DDH problem is intractable in $\mathbb{G}$, it follows that the discrete logarithm problem is also intractable in $\mathbb{G}$.

Symmetric encryption. We use a symmetric encryption scheme consisting of three polynomial time algorithms $(\mathsf{S.KeyGen}, \mathsf{S.Enc}, \mathsf{S.Dec})$. The $\mathsf{S.KeyGen}$ algorithm is used to generate the secret encryption key k. $\mathsf{S.Enc}$ is a deterministic algorithm that takes inputs as the key k and a message m and outputs a ciphertext c. The $\mathsf{S.Dec}$ algorithm takes input as the ciphertext c and outputs the plaintext of m using the key k.

Blockchain and Smart Contracts. Blockchain [33] is a distributed ledger technology (DLT). Multiple users can reach a consensus on the blockchain contents without a trusted central authority. The stored contents on the blockchain are also guaranteed to be tamper-proof. Smart contracts are some executable and turing-complete codes that can be deployed on the blockchain. After deployment, users will call smart contracts functions, blockchain nodes will execute the function codes and reach a consensus, and the results will be recorded to the blockchain as transactions.

C. Data Trading Process

Existing works generally conduct the data trading process in two phases: The first is for confirming the data availability, and the second is for data/payment exchange. In the first phase, the data buyer needs to make sure the data fits his/her needs and is indeed available for sale. In practice, different mechanisms are applied in this phase to meet specific requirements like confidentiality, crash tolerance, and authenticity. For example, in [13], a description (metadata) of data on sale is published on the blockchain as the commitment to data availability; In [16], the data would be fed directly to the smart contract as a sign of data availability; In [42], the authors use smart contract together with zero-knowledge proof to convince the buyer of both data retrievability and availability. After the buyer confirms the data matches his/her needs and is available, s/he will initiate the actual exchange phase, which includes the payment from the buyer and the data delivery from the seller. With the broad application of blockchain, the exchange phase now has natural transparency. Technologies like HTLC [36] can be applied to provide fair and atomic exchange between digital assets (like digital currency and data/keys to decrypt the ciphertext), which guarantees the security of data trading.

NiFT also applies the two-phase structure of data trading. During the correctness verification phase, similar to existing interactive fair exchange protocols, NiFT also followed the idea of a slicing and sampling process: the seller splits the whole dataset into slices and encrypts them; some slices will be randomly chosen and disclosed to the buyer (both the ciphertexts of the slices and corresponding decryption keys); the buyer can then verify the correctness of the data; and both parties will move on to the next phase if the correctness is validated, or otherwise, they will terminate the process.

During the atomic exchange phase, we also leveraged the idea of HTLC on blockchain to ensure the exchange between decryption keys and payments happens in an atomic way. As aforementioned, directly using the secret key as the input for the atomic exchange will inherently require the data to be transferred to the buyer secretly. Instead, we followed the idea of the seller secretly sharing some values first and then providing some other values as input for recovering the decryption keys [15], [20]. In Delgado et al.'s [20] work, however, the secret key retrieval is based on the vulnerability of Elliptic Curve Digital Signature Algorithms (ECDSA). Therefore, NiFT's atomic exchange phase is constructed based on FairTrade [15]. In their work, the buyer deploys a smart contract to the blockchain and locks some coins as a deposit. To claim the deposit, the seller must provide valid input, which can be used to recover the decryption keys. With the help of HTLC, when the seller provides a valid input, s/he can claim the deposit; meanwhile, when the buyer's payment is made, s/he can recover the decryption keys and then retrieve the data. The atomicity of the exchange is therefore achieved, and fairness is therefore ensured. Specifically, the seller first splits his/her secret key sk into two parts *s.t.* $sk = sk_1 + sk_2$; then the seller (privately) sends sk_1 to the buyer; the buyer will deploy an HTLC onto the blockchain; if the seller provides a correct sk_2 to the contract such that its sum with sk_1 equals to sk, then the seller will receive the deposit locked in the contract; otherwise, if time expires, the buyer will get the refund.

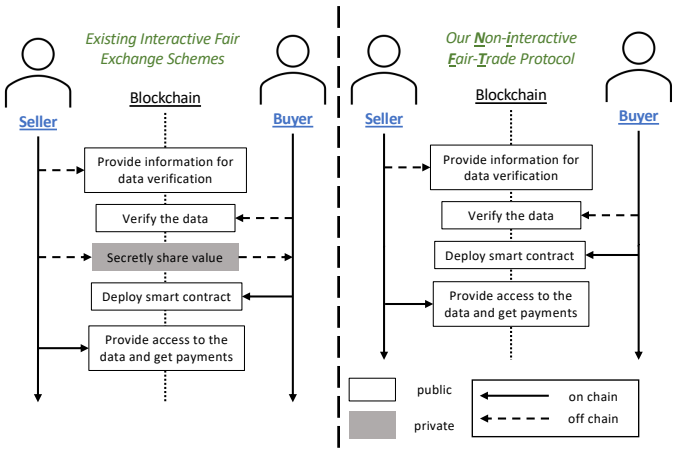

Fig. 1. General workflows of existing schemes (left) and NiFT (right).

A drawback of the existing solutions, however, is that to launch a smart contract, the seller has to send some value to the buyer in the first place, which leads to extra communication (as is shown in the left half in Figure 1). Moreover, since later, the smart contract's input is public, the value must be sent to the buyer privately. This further indicates that the seller has to build a secure channel with the buyer. We improved the previous protocol, where the buyer can deploy a smart contract without waiting for the seller's initial private sharing. Additionally, we also considered another possible misbehavior by an irrational user and provided a mitigation method.

IV. Our Design of NiFT

As is shown in the right part of Figure 1, the overall workflow of NiFT is as follows: the seller will broadcast some information to the blockchain for potential buyers to validate. Interested buyers could follow the provided information to validate the data (e.g., if the data size is large, the provided information could be a link that can be directed to where the data are stored). If the data passes the verification process, the buyer will deploy a smart contract on the blockchain, requiring the seller to provide valid decryption keys to the data (the reason why we use decryption keys to exchange with the payments was explained in Section III-A). Since the idea of HTLC is implemented when deploying the smart contract, atomicity is guaranteed such that the keys will be exchanged with the payments simultaneously. Note that NiFT mainly focuses on removing the requirements of a pair-wise secure channel between the buyer and the seller during the exchange phase. Therefore, we skip the discussion of the verification phase (which has been briefly explained in Section III-C).

A. Non-interactive Atomic Exchange without Pair-wise Secure Communication Channels

As described in Section III-C, we adopt the verification phase in existing approaches. Specifically, we also follow FairTrade's [15] design that uses symmetric encryption to protect the data for sale and uses a sampling mechanism to convince the buyer about the availability of the data. However, we applied a different atomic exchange phase after the buyer decided to buy the data in order to achieve the requirement of no pair-wise secure communication channels. Through the protocol analysis, we found two important attributes that ensure fairness and confidentiality during the exchange phase: first, the two parties need to secretly achieve an agreement/consensus on some value s, such that later on, this value can be used to generate and deploy the smart contract; second, the smart contract should be generated by the buyer and called by the seller such that the exchange phase can be completed. The second attribute is straightforward. For the first attribute, from the seller's side, since s/he also knows her/his secret key sk_S, s/he can check the smart contract's correctness and decide whether to provide a valid $input$ to the smart contract. On the other hand, from the buyer's point of view, since s/he only knows s, s/he cannot recover sk_S when the smart contract is deployed. However, as long as the seller provides the correct $input$, the buyer can immediately calculate sk_S, also indicating the atomicity is achieved by the protocol. Besides, outsiders (other users except the seller and the buyer) cannot retrieve sk_S because they only have access to the $input$ provided by the seller, which is insufficient. With these two points, the protocol can achieve fairness and confidentiality simultaneously.

However, the purpose of the first attribute is to reach an agreement on the value s between the two parties, which means the selection of s can be made by either of them. As a result, the protocol's security will remain the same if the buyer selects s and secretly reaches a consensus on its value with the seller. Additionally, as mentioned in the second attribute, the buyer must deploy the smart contract. Therefore, if the buyer can select s and secretly reach an agreement on the value s with the seller in the same smart contract, the extra communication round between the two parties can be saved. More importantly, the strict requirement for a secure channel between the seller and the buyer in the previous solution can also be further removed. Considering these ideas, we developed our novel design of the non-interactive atomic exchange without pair-wise secure communication channels as follows (shown in Figure 2).

In step 1, the buyer randomly chooses a secret number s, encrypts s with the seller's public key pk_S, and gets the ciphertext $c = A.Enc_{pk_S}(s)$. In step 2, the buyer generates a smart contract, including the ciphertext c, requiring that to claim the deposit in the contract, the seller must provide a valid $input$ such that $input = sk_S - s$. After the contract is deployed on the blockchain, in step 3, the seller will recover s by decrypting c with his/her secret key sk_S. Then, the seller will verify whether the contract's required value of $input$ is calculated correctly. The seller will terminate the process if the required $input$ value is invalid or provides the correct answer to claim the deposit (step 4). Here, we take ElGamal cryptosystem [23] as an example of our detailed implementation. The seller randomly chooses his/her secret key sk_S and calculates the corresponding public key, $pk_S = g^{sk_S}$, where g

4

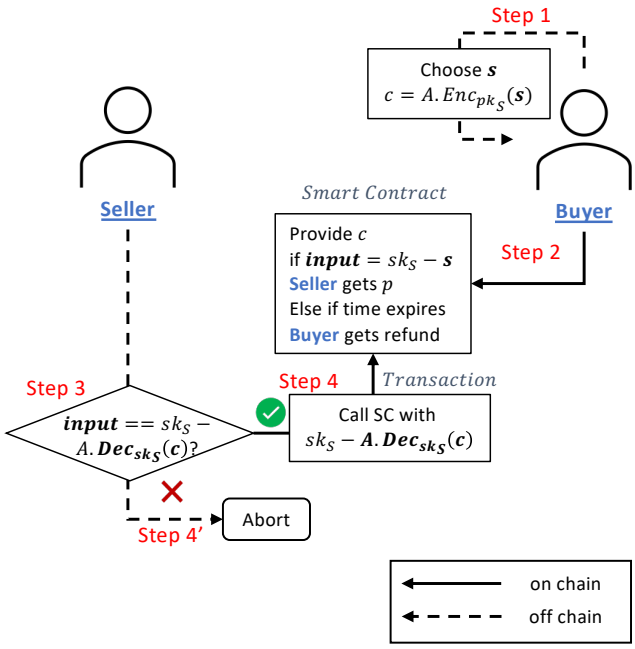

Fig. 2. The protocol of Non-interactive Atomic Exchange without Pair-wise Secure Communication Channels.

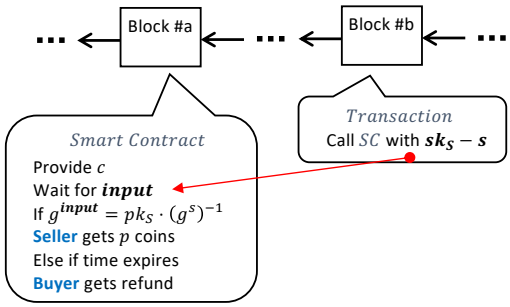

Fig. 3. An example of a smart contract call based on ElGamal.

is the generator of the group $\mathbb{G}$ in ElGamal. As is shown in Figure 3, the smart contract is deployed in Block #a, including the ciphertext c and a cryptographic puzzle based on the $input$ value. The puzzle requires that if $g^{input} = pk_S \cdot (g^s)^{-1}$, then the seller can redeem the locked p coins; otherwise, if time expires, the buyer will get a refund. Note that from the seller's view, after decryption, s/he can recover s. Since s/he also knows sk_S, therefore, if s/he calls the smart contract with $input = sk_S - s$, then $g^{input} = g^{sk_S - s} == g^{sk_S}/g^s$. Further, as $g^{sk_S} = pk_S$, then we will have $g^{input} = pk_S/g^s$, which further equals to $pk_S \cdot (g^s)^{-1}$.

B. Self-maintained Storage Service

Although our non-interactive atomic exchange protocol has removed the need for a pair-wise secure channel from the existing protocol, there might be another potential issue between the seller and previous buyers. In NiFT, after each successful transaction, one more buyer will have access to the data by recovering the seller's secret key sk_S. In other words, if sometime later a new buyer follows the protocol in Section IV-A, previous buyers can also decrypt c and provide a valid $input$. However, since only the buyer and the seller know the secret value in the previous work [15], outsiders (including previous buyers) cannot call the contract. Therefore, in NiFT, we further require sellers to maintain a storage service. A complete trading scenario and detailed description of each step can be found in Figure 4. Note that steps 1 to 4 are the same as those explained above in the exchange phase, so we skip discussing these steps.

At the beginning of the verification phase, the seller should broadcast some information with which potential buyers can access and validate the data (step 0.a). For instance, the seller could provide a link to a cloud/server where the data are stored if the data size is large. The information should include the encrypted data $S.Enc_{k_i}(d_i)$, the ciphertext of the symmetric keys $A.Enc_{pk_S}(k_i)$ and the selected sample of symmetric keys for verification purposes. If some buyer finishes the verification phase (step 0.b) and agrees to move on, both parties will continue to the exchange phase (steps 1 to 4). After they successfully complete the exchange phase, the seller will update part of the information provided in his/her storage service (step 5). Specifically, the seller will first choose another secret key sk_t, re-calculate its corresponding public key pk_t, and re-encrypt the symmetric keys $A.Enc_{pk_t}(k_i)$. Then, the seller will replace the previous public key and the ciphertext of symmetric keys (pk_S, $A.Enc_{pk_S}(k_i)$) with the new key and ciphertext.

According to the random sampling protocol in FairTrade [15], only the encrypted data $S.Enc_{k_i}(d_i)$ determined the random seed. This means updating the seller's public key and the ciphertext of symmetric keys will not affect the sampling process, further indicating the selected keys remain the same. In other words, after step 5, future buyers can still verify the data correctness with the updated information. We omit the exact verification phase as it is not our major contribution, which has been briefly introduced in Section III-C. With the self-maintained storage protocol, previous buyers cannot call future smart contracts as they cannot get the secret values encrypted by future buyers.

C. Security Analysis

In this section, we analyze the system security of NiFT. The major protocols in NiFT are based on public key encryption algorithms and certain properties will be used as building blocks when we prove our mechanism's security.

Lemma 1. *During the atomic exchange phase, it is impossible for any $\mathcal{S}$ to claim the deposit with a different value $input'$ such that $input' \neq input \pmod q$.*

Proof. The proof is straightforward based on the group theory. Suppose we have two integers $0 < x_1 \neq x_2 < q$ such that $g^{x_1} = g^{x_2} \pmod q$. Therefore, we will have $g^{x_1 - x_2} = 1$

5

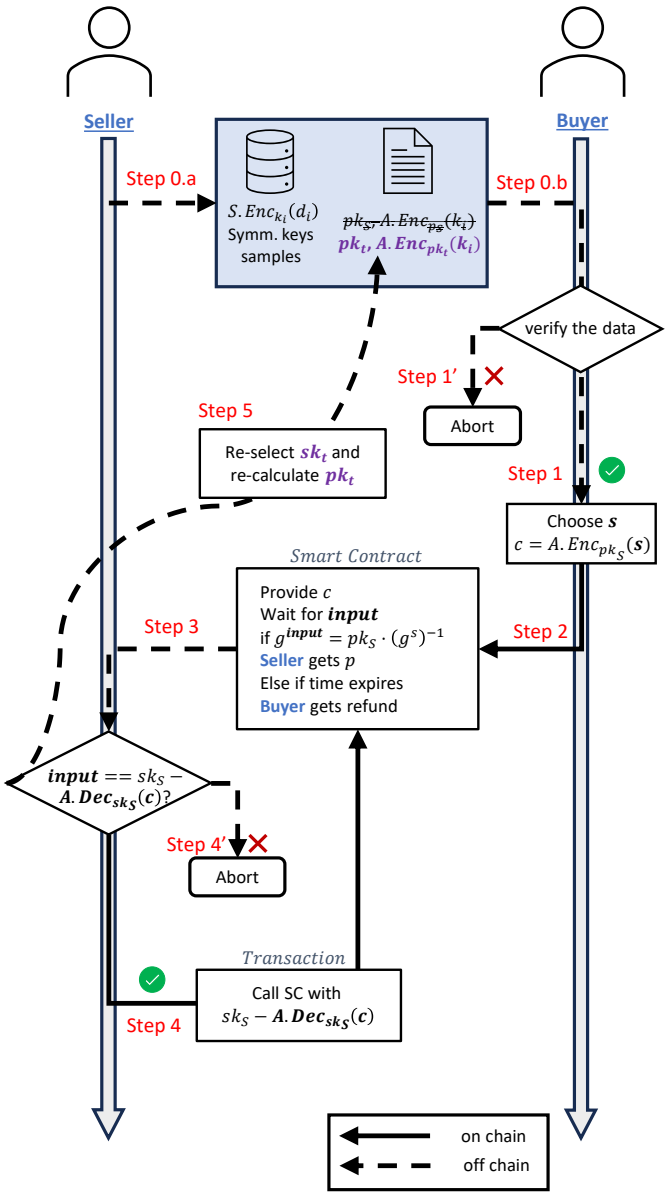

Fig. 4. The entire workflow details of NiFT.

(mod q). Since q is prime, we then have $g^{q-1} = 1 \pmod q$ due to the Little Fermat Theorem. That means we will have $x_1 - x_2 = k(q-1)$, where k is an integer. Since $0 < x_1 \neq x_2 < q$, we will have $g^{x_1} = g^{x_2} \pmod q$ *iff.* $k = 0$, meaning that $x_1 = x_2$, which is contradictory to our assumption. Therefore, it is impossible for $\mathcal{S}$ to claim the deposit with a different value $input'$ that is not congruent to the correct value $input$. □

Similarly, we have the following two lemmas:

Lemma 2. *Before the seller reveals the input, if the public key encryption key is cryptographically secure, it is hard for $\mathcal{B}$ to recover sk_S in polynomial time.*

Lemma 3. *During the whole exchange process, if the public key encryption key is cryptographically secure, it is hard for blockchain peers to recover sk_S in polynomial time.*

Proof. Both proofs are straightforward. In Lemma 2, before $input$ is provided, $\mathcal{B}$ only knows pk_S and s. To decrypt the data, $\mathcal{B}$ can recover either sk_S or $input$. Suppose there is a polynomial time algorithm $FindSK(pk_S)$ that outputs sk such that $pk_S = g^{sk}$, then $\mathcal{B}$ can access the data without paying to $\mathcal{S}$. Based on this assumption, given a group element $e \in \mathbb{G}$, we can use $FindSK(\cdot)$ to find d, such that $g^d = e$, in polynomial time, which also means $\mathcal{B}$ would have a polynomial algorithm to solve the discrete logarithm problem. Since the reduction is in polynomial time, it is at least as hard as solving the discrete logarithm problem for $\mathcal{B}$ to retrieve the secret key sk. Therefore, $\mathcal{B}$ cannot access the data unless s/he correctly deploys the smart contract and receives the valid $input$ from $\mathcal{S}$. Similar conclusions can also be made when the $\mathcal{B}$ wants to recover $input$ or blockchain peers in Lemma 3 want to recover sk_S or s. Therefore, it is at least as hard as solving the discrete logarithm problem for $\mathcal{B}$ or blockchain to recover sk_S without knowing $input$ or s. □

Theorem 1. *With Lemmas 1 to 3, we say that the design of NiFT can ensure trading fairness between $\mathcal{S}$ and $\mathcal{B}$, and prevent blockchain peers from inferring any secrets.*

Proof. We have proved that it is impossible for a malicious $\mathcal{S}$ to get the payment without giving a correct $input$ (in Lemma 1) and it is at least as hard as solving a discrete logarithm problem in polynomial time for a malicious $\mathcal{B}$ to access the data without paying to the seller (in Lemma 2). Furthermore, blockchain peers cannot infer the decryption key either, unless they can solve a discrete logarithm problem in polynomial time (in Lemma 3). Therefore, all three types of adversaries in Section III-A can be prevented by NiFT. □

Besides, we further explain why the seller needs to update the storage service each time an exchange is completed. As mentioned in Section IV-B, if the seller uses the same public/secret key pairs for every transaction, the previous buyers can call future contracts because they can recover the secret s with the same secret key. If a previous buyer provides a valid $input$ to some future smart contracts, s/he will be the seller in this case, and such a scenario can be treated as a re-selling process (and such a buyer is named a "re-seller" hereafter). Based on our assumption, both parties will not collude with each other, and at least one party is honest. Therefore, the re-seller will not provide a valid $input$ if the new contract is not correctly written. Meanwhile, the deposit cannot be spent if the re-seller cannot provide a correct $input$. According to our definition of fairness above, the exchange process itself is atomic and fair. However, it is the original seller who provides the storage service, so it is unfair to her/his side since s/he does not get paid. Additionally, the discussion of re-selling is not the major focus of this paper. Therefore, in NiFT, sellers are required to update their storage service

after each exchange is completed, and the re-selling scenario remains an open problem.

D. Discussion

We discuss specific design choices of NiFT.

No Private Off-chain Communication in Exchange. The biggest difference between our protocol and the FairTrade [15] is in the secret key exchange phase. While in FairTrade, the buyer and the seller need to communicate with each other to secretly share a random number through an off-chain secure channel before the actual trading on smart contract, NiFT only requires on-chain operations in the exchange phase. Even if the availability of storage services and availability to the same blockchain network is provided, the direct communication channel between two individuals may not exist by default or need extra process/infrastructure to establish. Eliminating off-chain communication reduces the likelihood of disputes arising from the delivery of the random number by the seller. Such disputes can occur when the seller and buyer make conflicting claims, thereby delaying the completion of the trading process. It also means that all communications between buyer and seller regarding the exchange are stored on-chain, which makes the exchange process auditable if needed [40].

Storage Service. In NiFT, the seller needs a storage service to host data and encrypted keys, which is also needed in previous works [15], [20] for the data verification phase. We claim such a service does not affect the decentralization of NiFT. Our protocol mainly focuses on the exchange part after the buyer has identified the availability and validity of the data. To guarantee the data are available for the buyer, the seller may use any cloud service or distributed storage service like IPFS [17] to avoid single-point-of-failure. The seller may also apply zero-knowledge proof to provide public verification ability to convince the buyer of the validity of the data on sale [39].

Limitations. Note that NiFT is not able to prevent non-auditable off-chain data trading/exchanging behaviors. Therefore, detecting/preventing collusion among buyers (who may share keys/data offline) is out of the scope. In fact, due to the nature of digital assets, it is hard to prevent malicious users from conducting offline copies.

V. Evaluation

The design of NiFT mainly focused on the exchange phase. Our non-interactive atomic exchange protocol is implemented on Ethereum, based on the idea of HTLC. Our smart contracts are written in Solidity and tested on Truffle for estimating the gas consumption of function calls. FairTrade is compared as a benchmark work, and Delgado's work [20] is also compared since they applied the two-phase structure as well. Moreover, we further analyzed and compared NiFT with other blockchain-based data trading protocols, such as the on-chain cost and the off-chain communication. Besides, as described in Section IV-B, sellers in NiFT are required to update their storage service after completing each transaction. Therefore, sellers in NiFT have an extra overhead of re-encrypting the symmetric keys. We also simulate this re-encryption process

TABLE I
Round-Trip Time (RTT) Under Different Transmission Distances (Wired Connection)

Distance	med. short	medium	med. long	long
avg (ms)	54.4	96.8	195.0	243.9

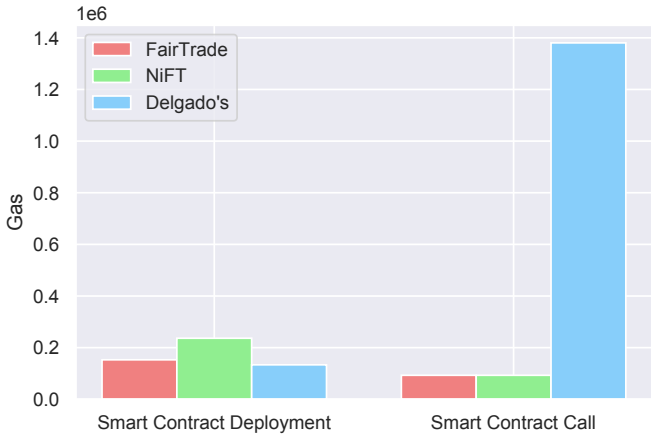

Fig. 5. Gas consumption for deploying and calling the exchange contract.

using some library in Python3, and the codes are running on a personal laptop with Intel Core i5 (2.3 GHz Quad-Core) with 16 GB memory. Our smart contracts and simulation codes are available on Anonymous GitHub [4].

A. Network Communication Latency Reduced by NiFT

We first estimate how much communication time can be saved by NiFT. As the distance between the geolocations will significantly influence the network latency, we tested some public servers provided by iPerf3 [26] and Table I shows the Round-Trip Time (RTT) for wired-connection with different servers, which is one of the measurements of the time needed for building secure channels (e.g., TLS handshake [3]). (Please kindly note that we cannot include the exact geolocations/distances in the table of any under-revision manuscript due to the double-blind policy.) We can observe that starting from a medium distance, the RTT would be close to 100ms, which indicates a potential revenue loss begins [38]. Note that the latency may be further influenced by the seller's individual network resources, e.g., bandwith and consistency.

B. Gas Consumption During the Atomic Exchange Phase

Figure 5 shows the gas consumption for both the smart contract deployment and the smart contract call. We can see that the buyer in NiFT spends more gas to deploy the smart contract onto the blockchain. The extra consumption for smart contract deployment is 83531 gas more than the benchmark work, which is around $0.18 USD as of December 2023 [2]. However, such overhead is only on the buyer's side, and since both parties can benefit from the reduced communication, the buyer may further negotiate the price with the seller. On the other hand, same amount of gas is consumed when the seller calls the smart contract as the benchmark work. Although in NiFT, the seller needs extra decryption on c

TABLE II
COMPARISON AGAINST OTHER PROTOCOLS

	Data Matching Process	Trading Object	On-chain Cost	Off-chain Communication
Data Analytic Trading [16]	A trusted broker will match the buyer and data for sale	Analysis results	Smart contract for data analysis	Requires secure communications between trusted nodes
Optimistic Trading [21], [22]	Optimistic trade with post-deal dispute resolving	Data set	HTLC-based judge for dispute resolving	Requires a pair-wise secure channel for data transmission
Delgado's [20]	Buyer-request sampling verification	Data set	HTLC	Requires a pair-wise secure channel for secret value
FairTrade [15]	Random sampling-based verification	Data set	HTLC	Requires a pair-wise secure channel for secret value
NiFT	Random sampling-based verification	Data set	HTLC	No pairwise off-chain communication for secret value

TABLE III
TIME COST FOR SYMMETRIC KEYS RE-ENCRYPTION

Number of symm. keys	1000	2000	3000	4000
Time cost of re-enc. (s)	87.6	144.7	214.1	339.1

and calculation on $input$, both processes are done locally, which will not cause extra on-chain costs. Besides, we also found that buyers in both NiFT and FairTrade spend more gas than those in Delgado's work when deploying smart contracts. This is because, in Delgado's work, the authors leveraged the vulnerability of ECDSA (Elliptic Curve Digital Signature Algorithm) for the atomic exchange, where if one uses the same random number to generate two signatures, others can infer the secret key. Therefore, the only required entry for buyers in their work is to ensure the random numbers are the same. However, more gas is consumed when sellers call the contracts because the signature verification and other calculations are resource-consuming.

C. Seller's Overhead for Self-maintained Storage Service

In Table III, we present the simulation results for the seller to update the symmetric keys' ciphertext after each transaction. Note that we did not include the time for the public key's recalculation as it is negligible (less than 0.1s on average) compared to the re-encryption process. Even though the time required for the re-encryption process increases linearly to the number of keys, the seller can still finish the process locally/off-chain, which will not cost any on-chain resources. Moreover, the seller does not need to update once each time a transaction is finished. Instead, the seller can prepare many public keys and their re-encryption results in advance to simply fetch and upload them after each transaction.

D. Comparison With other Interactive Fair Exchange Schemes

We also compare NiFT and other blockchain-based data trading protocols in Table II. Some projects focus on trading the result of analysis (a given algorithm output on a data set, like [16]) instead of the data set itself. These projects usually require additional trust assumptions (like Trusted Execution Environment) besides the basic blockchain and smart contract due to the additional privacy concern of running analysis over private data sets. These projects are usually hard to deploy. Data trading applying optimistic trading modes like [21], [22] utilize blockchain and smart contract as a judge for dispute resolving. These works usually assume a successful exchange in the first place (which is therefore considered optimistic), and a follow-up protocol (e.g., the judge contract) will be called later if some disputes appear. The multi-round process of presenting proofs of misbehavior will involve too much on-chain cost and does not scale well, which may limit their application scenario. Data trading like [15], [20] and NiFT deploy a sampling-based verification phase, which allows buyers to get a small portion of data to check if the data set matches their needs. Then, their trading process will use HTLC to exchange the payments and the key to decrypt the whole data set. The downside of these two projects is they require off-chain communication to initialize the HTLC exchange, which increases the possible interface for dispute and, hence, a higher chance of exchange stalling/delay. While in NiFT, the off-chain communication is removed, which makes the exchange only depend on the status of the HTLC execution.

VI. CONCLUSION

In the rapidly growing field of data trading, traditional methods relying on a Trusted Third Party (TTP) are becoming obsolete. Blockchain technology offers new avenues for ensuring fair trading between untrusted parties but often requires impractical secure channels for communication. In this paper, we present NiFT, a non-interactive fair trading protocol that eliminates the need for direct secure channels between the buyer and the seller. By integrating the secret-sharing process into smart contracts, NiFT ensures a fair and secure exchange purely on-chain, thereby avoiding off-chain communication disputes. Additionally, our protocol prevents unauthorized users from claiming buyers' deposits and incurs minimal on-chain overhead. Security proofs and simulation results further validate the efficacy and safety of NiFT in addressing the challenges of fair data trading.

REFERENCES

[1] Data marketplace platform market size; share report, 2030, https://www.grandviewresearch.com/industry-analysis/data-marketplace-market-report.
[2] Gwei to us dollar, https://www.cryps.info/en/Gwei_to_USD/.
[3] The transport layer security (tls) protocol version 1.3 , https://www.rfc-editor.org/rfc/pdfrfc/rfc8446.txt.pdf.
[4] https://anonymous.4open.science/r/NiFT-6C06/README.md.

[5] Nadarajah Asokan, Victor Shoup, and Michael Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186)*, pages 86–99. IEEE, 1998.

[6] Feng Bao, Robert H Deng, and Wenbo Mao. Efficient and practical fair exchange protocols with off-line ttp. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186)*, pages 77–85. IEEE, 1998.

[7] Debopam Bhattacherjee, Waqar Aqeel, Sangeetha Abdu Jyothi, Ilker Nadi Bozkurt, William Sentosa, Muhammad Tirmazi, Anthony Aguirre, Balakrishnan Chandrasekaran, P Brighten Godfrey, Gregory Laughlin, et al. {cISP}: A {Speed-of-Light} internet service provider. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 1115–1133, 2022.

[8] Manuel Blum. How to exchange (secret) keys. *AcM Transactions on computer systems (Tocs)*, 1(2):175–193, 1983.

[9] Philip Lewis Bohannon. Transport layer security latency mitigation, October 10 2017. US Patent 9,787,643.

[10] Dan Boneh. The decision diffie-hellman problem. In *International Algorithmic Number Theory Symposium*, pages 48–63. Springer, 1998.

[11] Michael Borkowski, Daniel McDonald, Christoph Ritzer, and Stefan Schulte. Towards atomic cross-chain token transfers: State of the art and open questions within tast. *Distributed Systems Group TU Wien (Technische Universit at Wien), Report*, 2018.

[12] Michael Borkowski, Christoph Ritzer, Daniel McDonald, and Stefan Schulte. Caught in chains: claim-first transactions for cross-blockchain asset transfers. *TU Wien: Technische Universität Wien, Tech. Rep*, 2018.

[13] Chuan Chen, Jiajing Wu, Hui Lin, Wuhui Chen, and Zibin Zheng. A secure and efficient blockchain-based data trading approach for internet of vehicles. *IEEE Transactions on Vehicular Technology*, 68(9):9110–9121, 2019.

[14] Zhili Chen, Wei Ding, Yan Xu, Miaomiao Tian, and Hong Zhong. Fair auctioning and trading framework for cloud virtual machines based on blockchain. *Computer Communications*, 171:89–98, 2021.

[15] Changhao Chenli, Wenyi Tang, and Taeho Jung. Fairtrade: Efficient atomic exchange-based fair exchange protocol for digital data trading. In *2021 IEEE International Conference on Blockchain (Blockchain)*, pages 38–46. IEEE, 2021.

[16] Weiqi Dai, Chunkai Dai, Kim-Kwang Raymond Choo, Changze Cui, Deiqing Zou, and Hai Jin. Sdte: A secure blockchain-based data trading ecosystem. *IEEE Transactions on Information Forensics and Security*, 15:725–737, 2019.

[17] Erik Daniel and Florian Tschorsch. Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks. *IEEE Communications Surveys & Tutorials*, 24(1):31–52, 2022.

[18] Martijn de Vos, Can Umut Ileri, and Johan Pouwelse. Xchange: A universal mechanism for asset exchange between permissioned blockchains. *World Wide Web*, pages 1–38, 2021.

[19] Sergi Delgado-Segura, Cristina Pérez-Solà, Jordi Herrera-Joancomartí, and Guillermo Navarro-Arribas. Bitcoin private key locked transactions. *Information Processing Letters*, 140:37–41, 2018.

[20] Sergi Delgado-Segura, Cristina Pérez-Solà, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. A fair protocol for data trading based on bitcoin transactions. *Future Generation Computer Systems*, 107:832–840, 2020.

[21] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 967–984, 2018.

[22] Lisa Eckey, Sebastian Faust, and Benjamin Schlosser. Optiswap: Fast optimistic fair exchange. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 543–557, 2020.

[23] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[24] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[25] Juntao Gao, Tong Wu, and Xuelian Li. Secure, fair and instant data trading scheme based on bitcoin. *Journal of Information Security and Applications*, 53:102511, 2020.

[26] Vivien GUEANT. Iperf - the ultimate speed test tool for tcp, udp and sctptest the limits of your network + internet neutrality test, https://iperf.fr/iperf-servers.php#public-servers.

[27] Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM symposium on principles of distributed computing*, pages 245–254, 2018.

[28] Yan Li, Lingyan Li, Yanqi Zhao, Nadra Guizani, Yong Yu, and Xiaojiang Du. Toward decentralized fair data trading based on blockchain. *IEEE Network*, 35(1):304–310, 2020.

[29] Sangwon Lim, Hyeonmin Lee, Hyunsoo Kim, Hyunwoo Lee, and Taekyoung Kwon. Ztls: A dns-based approach to zero round trip delay in tls handshake. In *Proceedings of the ACM Web Conference 2023*, pages 2360–2370, 2023.

[30] James A Liu. Atomic swaptions: cryptocurrency derivatives. *arXiv preprint arXiv:1807.08644*, 2018.

[31] Jian Liu, Wenting Li, Ghassan O Karame, and N Asokan. Toward fairness of cryptocurrency payments. *IEEE Security & Privacy*, 16(3):81–89, 2018.

[32] Steve Lohr. For impatient web users, an eye blink is just too long to wait, https://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html, Mar 2012.

[33] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.

[34] Henning Pagnia and Felix C Gärtner. On the impossibility of fair exchange without a trusted third party. Technical report, Technical Report TUD-BS-1999-02, Darmstadt University of Technology ..., 1999.

[35] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *International Journal of Information Security*, 16(1):1–22, 2017.

[36] R Russell. Lightning networks part ii: Hashed timelock contracts (htlcs). *See https://rusty. ozlabs. org*, 2015.

[37] Joseph Salowey, Hao Zhou, Pasi Eronen, and Hannes Tschofenig. Transport layer security (tls) session resumption without server-side state. Technical report, 2008.

[38] Ankit Singla, Balakrishnan Chandrasekaran, P Brighten Godfrey, and Bruce Maggs. The internet at the speed of light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2014.

[39] Xiaoqiang Sun, F Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205, 2021.

[40] Shigeya Suzuki and Jun Murai. Blockchain as an audit-able communication channel. In *2017 IEEE 41st annual computer software and applications conference (COMPSAC)*, volume 2, pages 516–522. IEEE, 2017.

[41] Wei Xiong and Li Xiong. Data resource protection based on smart contract. *Computers & Security*, 98:102004, 2020.

[42] Liang Xue, Jianbing Ni, Dongxiao Liu, Xiaodong Lin, and Xuemin Shen. Blockchain-based fair and fine-grained data trading with privacy preservation. *IEEE Transactions on Computers*, 2023.

[43] Yanqi Zhao, Yong Yu, Yannan Li, Gang Han, and Xiaojiang Du. Machine learning based privacy-preserving fair data trading in big data market. *Information Sciences*, 478:449–460, 2019.