SCIENCE CHINA
Information Sciences

July 2022, Vol. 65 170308:1–170308:2
https://doi.org/10.1007/s11432-021-3441-y

• LETTER •
Special Focus on Cyber Security in the Era of Artificial Intelligence

Post quantum secure fair data trading with deterability based on machine learning

Jinhui LIU[1,2], Yong YU[3*], Hongliang BI[1], Yanqi ZHAO[3], Shijia WANG[4] & Huanguo ZHANG[5]

[1]*School of Cybersecurity, Northwestern polytechnical University, Xi'an* 710072*, China;*
[2]*Research & Development Institute of Northwestern Polytechnical University, Shenzhen* 518057*, China;*
[3]*School of Cybersecurity, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China;*
[4]*School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin* 300071*, China;*
[5]*School of Cyber Science and Engineering, Wuhan University, Wuhan* 430072*, China*

Received 1 September 2021/Revised 19 January 2022/Accepted 1 March 2022/Published online 23 June 2022

Citation Liu J H, Yu Y, Bi H L, et al. Post quantum secure fair data trading with deterability based on machine learning. Sci China Inf Sci, 2022, 65(7): 170308, https://doi.org/10.1007/s11432-021-3441-y

Dear editor,

With the development of the Internet and digitalization, databases and data-analysis tools (for example, artificial intelligence and machine learning) have exhibited a significant potential to improve service quality. Data have become an invaluable asset, which is the basis of new data-training models [1]. A conventional data-trading ecosystem comprises three parties: seller (data owner), buyer, and data-exchange platform (third party). The seller sends the dataset to the data-exchange platform and sets an appropriate selling price; the buyer selects a data product and places an online order. Upon receiving the buyer's payment, the data-exchange platform transmits the purchased data to the buyer and pays the seller. In the system, the data-exchange platform acts as a trusted third party, participates in the total exchange, and matches the buyer's requirements. The platform may receive expensive charges and experience server downtime and single-point failure. Therefore, there is an urgent need to realize fair data exchange with decentralization. First, we propose a new cryptographic-primitive-designated verifier for double authentication, preventing signature (DVDAPS), in which only a designated verifier can verify the signature and the signer is punished once he commits a mistake. Additionally, we provide a concrete lattice-based construction method. Subsequently, we provide post-quantum-secure fair data trading with deterability based on machine learning. Some theorems on security analysis show that the proposed fair data-trading protocol satisfies the properties of completeness, fairness, accountability, and privacy in the random-oracle model.

Lattice based DVDAPS using QAP. Let a QAP for circuit C be $\mathrm{QAP}(C) \to \mathcal{Q}(\mathcal{A},\mathcal{B},\mathcal{C},t(x))$, where $\mathcal{A}=\{a_i(x)\}_{i=0}^{l}$, $\mathcal{B}=\{b_i(x)\}_{i=0}^{l}$, $\mathcal{C}=\{c_i(x)\}_{i=0}^{l}$ and a polynomial $t(x)$ such that $\deg(t(x)) \geqslant \max\{\deg(a_i(x)),\deg(b_i(x)),\deg(c_i(x))\}$ for all $i\in[0,l]$.

$\mathtt{DkeyGen}(1^\lambda)$. Pick a private key $\boldsymbol{s}\leftarrow \boldsymbol{Z}_q^n$, a key value binding PRF $\mathcal{F}: S\times D\to R$ with respect to $\hat{\beta}\in D$, and output $\mathrm{pk}:=(\mathtt{Enc}(\alpha),\mathtt{Enc}(\beta),\mathtt{Enc}(\delta),\{\mathtt{Enc}(s^i)\}_{i=0}^{d}, \{\mathtt{Enc}(\boldsymbol{s},0)\}_j,\{\frac{\beta a_i(s)+\alpha b_i(s)+c_i(s)}{\delta}\}_{i\in I_{mid}},\{\mathtt{Enc}(\frac{s^i t(s)}{\delta})\}_{i=0}^{d}, \hat{\boldsymbol{b}})$ and $\mathrm{sk}:=(\mathrm{sk}_{\mathtt{PRF}},\alpha,\beta,\delta,\boldsymbol{s})$, where $\alpha,\beta,\delta,s\leftarrow \boldsymbol{Z}_p$.

$\mathtt{Dsign}(\mathrm{sk},m_0,m_1)$. This algorithm takes random numbers $\gamma_a,\gamma_b\leftarrow \boldsymbol{Z}_p$ and computes $\hat{z}=\rho m_1+\boldsymbol{s} \bmod q$, $\hat{A}:=\mathtt{Enc}(A(s)),\hat{B}:=\mathtt{Enc}(B(s)),\hat{C}:=\mathtt{Enc}(C(s))$, where $A(s)=\alpha+\sum_{i=0}^{l}d_i a_i(s)+\gamma_a\delta$, $B(s)=\beta+\sum_{i=0}^{l}d_i b_i(s)+\gamma_b\delta$ and $C(s)=\frac{\sum_{i\in I_{\mathrm{mid}}}d_i(\beta a_i(s)+\alpha b_i(s)+c_i(s))}{\delta}+\gamma_a A(s)+\gamma_b B(s)-\gamma_a\gamma_b\delta$.

The defined proof relation $\mathcal{R}$ is $((\hat{\beta},\hat{c},m_0,\hat{z}_1,\hat{\boldsymbol{b}},A,B,C),(\mathrm{sk}_{\mathtt{PRF}},\rho,\boldsymbol{s},A(s),B(s),C(s)))\in\mathcal{R}\Leftrightarrow \rho=\mathcal{F}(\mathrm{sk}_{\mathtt{PRF}},m_0)\wedge\hat{z}=\rho m_1+\boldsymbol{s}\wedge\hat{c}=\mathcal{F}(\mathrm{sk}_{\mathtt{PRF}},\hat{\beta})\wedge\hat{\boldsymbol{b}}=f(\boldsymbol{s})=\langle\boldsymbol{a},\boldsymbol{s}\rangle+\boldsymbol{y}+\frac{q}{p}\boldsymbol{x}\wedge\hat{A}:=\mathtt{Enc}(A(s))\wedge\hat{B}:=\mathtt{Enc}(B(s))\wedge\hat{C}:=\mathtt{Enc}(C(s))$. Here, $\boldsymbol{x}=\{A(s),B(s),C(s),\alpha,\beta,\delta,s^i,\ldots\}$ and $\boldsymbol{y}\leftarrow\chi_\sigma$. Compute $(d_1,\ldots,d_l)$ for the value of each wire and $h(x)$ satisfying $(\sum_{i=0}^{l}d_i a_i(x))(\sum_{i=0}^{l}d_i b_i(x))-(\sum_{i=0}^{l}d_i c_i(x))=h(x)t(x)$ and $d_0=0$, and output a signature $\sigma:=(\hat{z},\pi)$, where $\pi=(\hat{A},\hat{B},\hat{C})$.

$\mathtt{Dver}(\mathrm{pk},\alpha,\beta,\delta,s,\boldsymbol{s},\sigma)$ decrypts σ to obtain $\mathtt{Dec}(\sigma,\boldsymbol{s})=(a,b,c)$, and verifies if the following equation holds. $a\cdot b=\alpha\cdot\beta+\sum_{i\in I_{\mathrm{public}}}d_i(\beta a_i(s)+\alpha b_i(s)+c_i(s))+c\delta$.

$\mathtt{Dextract}(\mathrm{pk},m_0,m_1,m_2,\sigma_1,\sigma_2)$. On the input message pair (m_0,m_1) and (m_0,m_2) and signature pair σ_1,σ_2, the algorithm computes $\boldsymbol{s}=\frac{m_1\hat{z}_2-m_2\hat{z}_1}{m_1-m_2}$ if some disputes exist.

The protocol construction. Figure 1 shows the main procedure of the proposed fair data-trading protocol based on DVDAPS and machine learning (ML). In blockchain transactions, the input script is a signature on the sender's secret key and the output script is a signature verification with the receiver's public key. σ_X is an X' signature on the body of $[T_x]$, where T_x (in: T_y), in: σ_X, out(body, σ): $\mathrm{ver}_Y(\mathrm{body},\sigma)$, value v, and lock time t.

* Corresponding author (email: yuyongxy@163.com)

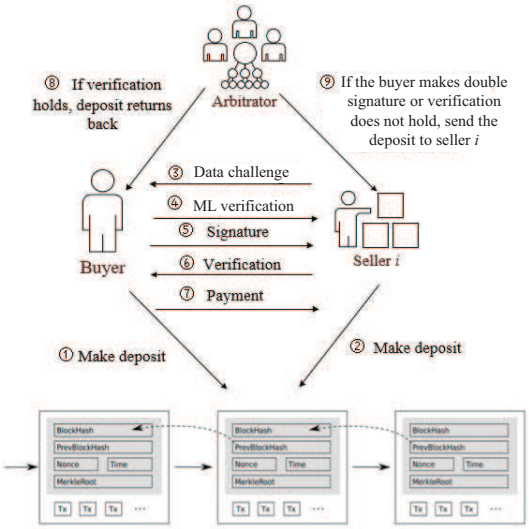

Figure 1 A framework of our fair data trading.

Setup. Suppose that the buyer and the arbitrator have a key pair $X = (\mathrm{PK}_X, \mathrm{SK}_X)$ respectively, where $X \in \{B, A\}$, and suppose seller i has a key pair $S_i = (\mathrm{PK}_{S_i}, \mathrm{SK}_{S_i})$ in the bitcoin system, where $i \in \{1, 2, \ldots, n\}$. Let $\mathtt{Dsign}_X(M)$ be the signature of data M using secret key SK_X of party $X \in \{A, S_i\}$.

Buyer B prepares to buy data instantiation F_i from seller S_i using d bitcoins. B builds a bitcoin contract to produce a common view with S_i. First, both B and S_i commit d' bitcoins as deposits, where $d' \geqslant d$. Subsequently, S_i receives d' bitcoins as compensation if B does not pay d bitcoins after S_i completes the verification of the algorithm. If S_i does not have data instantiation F_i, B obtains d' bitcoins as a compensation.

Contract building. B creates a transaction T_y that can only be redeemed using the secret key of S_i.

T_D(in : T_y); ins: $\mathtt{Dsign}_B([T_D])$, out (body, σ_1, σ_2): $(\mathtt{Dver}_B(\mathrm{body}, \sigma_1) \wedge \mathtt{Dver}_{S_i}(\mathrm{body}, \sigma_2)) \vee (\mathtt{Dver}_B(\mathrm{body}, \sigma_1) \wedge \mathtt{Dver}_A(\mathrm{body}, \sigma_2))$; value: d'; lock time: 0.

S_i creates a new transaction T_D, and sends it to the ledger. Subsequently, B builds the bodies of transaction T_p, signs the transaction, and sends $[T_P]$ to S_i. If T_D does not appear on the ledger and T_P does not provide for S_i in time t_1, then he aborts.

T_P(in : T_D); ins: $\mathtt{Dsign}_B([T_P]), \mathtt{Dsign}_{S_i}([T_P]), \perp$; out (body, σ): $\mathtt{Dver}_{S_i}(\mathrm{body}, \sigma)$; value: d'; lock time: t.

First S_i makes data challenge to show that he has trading data. B sends data instantiation F_i to seller S_i to show that he wants to buy data. The seller matches data instantiation F_i according to a machine-learning algorithm. If the matching holds, B computes signature $\sigma := (\hat{z}, \pi)$ using the signature algorithm in `DVDAPS` and sends it to S_i.

Verification and payment. S_i verifies σ and B offers d bitcoins to S_i and gets his deposit back. There are three phases in this process.

- In phase 1, B rejects to pay d bitcoins to S_i after he gets the results in deadline time.
- In phase 2, S_i asks A to output an abort, waits until deadline time, and obtains d' bitcoins deposit of B.
- In phase 3, if B does not send the signature σ to S_i or refuses to work together with B to retrieve his deposit or B makes double signatures, S_i can obtain the deposit of B with the help of A.

In the following, we introduce the three phases in detail.

Phase 1. (1) B sends a signature σ to S_i before time t_2. S_i goes to phase 3 if it does not receive σ.

(2) If σ is a valid signature, B creates new transaction T_{pay} and sends it to the ledger. S_i obtains d bitcoins as a reward.

T_{pay} (in: $T_{y'}$), in: $\mathtt{Dsign}_B([T_{\mathrm{pay}}])$, out (body, σ): $\mathtt{Dver}_{S_i}(\mathrm{body}, \sigma)$, value: d, lock time: 0.

(3) B builds transaction T_G, signs the transaction, and sends the signed body of T_G to S_i. Finally, B obtains its deposit.

T_G (in: T_D); in: $\mathtt{Dsign}_B([T_G])$ and $\mathtt{Dsign}_{S_i}([T_G])$; out (body, σ): $\mathtt{Dver}_B(\mathrm{body}, \sigma)$; value: d'; lock time: 0.

Phase 2. If B does not pay d bitcoins to S_i before t_3.

(1) S_i connects with A. S_i calculates signature $\mathtt{Sign}(\mathrm{SK}_{S_i}, \mathrm{abort} \| \sigma)$, and sends it to A.

(2) The payment protocol is terminated when time t_2 is not met. If the verification is successful and phase 3 has not been performed, A brings abort to the public and sends σ to B, where $\mathrm{TK} = \mathtt{Sign}(\mathrm{SK}_A, \mathtt{Sign}(\mathrm{SK}_{S_i}, \mathrm{abort}))$.

(3) B waits until time t, transfers items T_p, and obtains d' bitcoins from S_i as compensation.

Phase 3. This phase is ran by the three parties B, S_i, A. If B makes a double signature on colliding messages, S_i cannot receive the signature σ at time t_1, and B neither operates phase 2 nor signs the transaction body T_G in time t_4.

(1) S_i connects with A and sends a time-stamped transcript and tamper-proof proof to A. S_i calculates $\mathtt{Sign}(\mathrm{SK}_{S_i}, \mathrm{proof})$ and transmits them to A.

(2) A verifies the signature. If the verification algorithm holds, two cases must be considered.

If time goes beyond time t_3, A connects with B and requests him to offer the verification result. If B sends the signature to A, A rejects the request of B and sends the results to him/her. Otherwise, A signs the transaction T_{get} using secret key SK_A and provides the signed body of T_{get} to B. B transfers the transaction T_{get} and obtains the d' bitcoin deposit.

If the current time goes beyond time t_4 and B does not execute phase 2, A sends the signature body of T_{get} to B, and B receives d' bitcoin deposits.

T_{get}(in: T_D); in: $\mathtt{Dsign}_B([T_{\mathrm{get}}])$, $\mathtt{Dsign}_A([T_{\mathrm{get}}])$, out (body, σ), and $\mathtt{Dver}_B(\mathrm{body}, \sigma)$; val: d'; lock time: 0.

The security proof of the proposed `DVDAPS` scheme and the constructed protocol can be found in Appendix B.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61802239, 61872229, U19B2021, 12101333, 62074131), Key Research and Development Program of Shaanxi Province (Grant Nos. 2020ZDLGY09-06, 2020ZDLGY06-04, 2021ZDLGY06-04, 2021ZDLGY05-01), and Shenzhen Fundamental Research Program (Grant No. 20210317191843003).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zheng S, Pan L, Hu D, et al. A blockchain-based trading platform for big data. In: Proceedings of IEEE Conference on Computer Communications Workshops, 2020. 991–996