Secure Exchange of Digital Goods in a Decentralized Data Marketplace

Ariel Futoransky
Wibson & Disarmista

Carlos Sarraute
Wibson

Ariel Waissbein
Wibson & Disarmista

Daniel Fernandez
Wibson

Matias Travizano
Wibson

Martin Minnoni
Wibson

Abstract

We are tackling the problem of trading real-world private information using only cryptographic protocols and a public blockchain to guarantee honest transactions. In this project, we consider three types of agents —buyers, sellers and notaries— interacting in a decentralized privacy-preserving data marketplace (dPDM) such as the Wibson data marketplace. This framework offers infrastructure and financial incentives for individuals to securely sell personal information while preserving personal privacy. Here we provide an efficient cryptographic primitive for the secure exchange of data in a dPDM, which occurs as an atomic operation wherein the data buyer gets access to the data and the data seller gets paid simultaneously.

1 Preliminaries

In this paper, we are interested in the problem of trading real-world private information using only cryptographic protocols and a public blockchain to guarantee honest transactions. Private information in this context refers to attributes or events associated with a single individual (person or organization).

In [1] the authors introduced the notion of a decentralized Privacy-Preserving Data Marketplace (dPDM). A decentralized Data Marketplace (dDM) is a Data Marketplace (DM) with no central authority, no central data repository and no central funds repository. Additionally, a dPDM allows users to sell private information, while providing them privacy guarantees such as:

- Participants anonymity: the identities of the Sellers and Buyers are not revealed without their consent. In particular, the identity of the Data Seller is not revealed to the Data Buyer, without the consent of the Data Seller.
- Transparency over Data usage: the Data Seller always has visibility on how his Data is used by the Buyer.

Here we consider three types of agents interacting in a dPDM both privately (through end-to-end communications) and publicly through a permissionless blockchain:

Data Seller: The Seller $\mathcal{S}$ is the owner and subject of the private information that will be traded. He decides when and if his data is sold.

Data Buyer: The Buyer $\mathcal{B}$ is interested in acquiring information from Sellers, provided that the information meets the Buyer's quality requirements.

We expect Data Buyers to be organizations that will receive information from willing and actively participating individuals to train Data Science algorithms and models, with the benefit of knowing that the personal information should be accurate and current.

Notary: The Notary $\mathcal{N}$ has the means to validate information associated with Sellers. He is trusted by Buyers to certify the precision and quality of the data traded. The Notary is the only public player with a formal track record and a public reputation.

The information traded is collected mainly outside of the blockchain. The Notary will typically access the data as part of its business operations with the Sellers. The Buyer understands the value associated with the privileged position of the Notary and knows about the incentives aligned with its reputation.

We illustrate the three market roles with an example. Suppose that the Seller is a client of a Bank, who offers on the market his (anonymized) credit card transactions. The Buyer can be any entity requiring transactional data to train its Machine Learning models. In this example, the Bank is the ideal Notary since:

- The Bank can verify that the Seller is actually a client of the Bank, by requiring the Seller to provide information that authenticates him/her.
- The Bank can act as a Notary in case of conflict, and verify whether the information of credit card transactions sent by the Seller to the Buyer is valid and trustworthy (in particular, by comparing with the Bank's own records of the client's credit card transactions).

2 Problem Statement and Related Work

We assume that the Data Seller $\mathcal{S}$ and Data Buyer $\mathcal{B}$ are participating in a Data Exchange protocol, such as the Wibson protocol [1], and have already completed the following steps:

- The Buyer $\mathcal{B}$ has verified that the Seller $\mathcal{S}$ belongs to the Buyer's audience of interest.
- The Data requested is available.
- Buyer $\mathcal{B}$ and Seller $\mathcal{S}$ have agreed on a price that is acceptable to both parties.

They are now faced with the following challenges:

(1) If the Seller releases the information first, the Buyer may decide not to pay.

(2) If the Buyer pays first, the Seller may decide not to reveal the information.

(3) The Seller may reveal incomplete or false information.

Challenges (1) and (2) are known as the problem of **fair exchange**, which has been studied for decades. Study [2] showed that fairness is unachievable without the aid of a trusted third party. However, the blockchain can fill the role of the trusted party, and essentially eliminates the trust problem.

Zero Knowledge Contingent Payment (ZKCP) protocols have been proposed, which allow the fair exchange of goods and payments over the Bitcoin network [3]. The ZKCP protocols require the execution of a zero-knowledge proof in order to work. Interesting applications of the zero-knowledge proof of binding provided by ZKCP include the query to a database of passwords and hashes [4], a problem tackled from a Private Information Retrieval perspective in [5].

When ZKCP was first introduced in 2011 it was only theoretical as there were no known efficient zero-knowledge protocols that could be used for the proofs at that time. Since then, advances have been made and there are now general-purpose Succinct Non-Interactive Arguments of Knowledge (zk-SNARK) protocols that allow the implementation of the necessary proofs [6, 7].

However, the zk-SNARK protocols are expensive in terms of computational and transactional cost required to execute them on a blockchain such as Ethereum. Here we propose a solution that leverages the trust that the Buyer places in the Notary to solve Challenge (3) (ensure data quality) as well as Challenges (1) and (2) (fair exchange) in a very efficient way.

3 Secure Exchange of Digital Goods

3.1 Protocol SEDG1

The first contribution of this paper is the protocol described in Table 1, which enables a "Secure Exchange of Digital Goods" (SEDG) by leveraging the Buyer's trust in the Notary in order to solve the challenges previously mentioned.

Table 1: Secure Exchange of Digital Goods (protocol 1)

Notary $\mathcal{N}$	Seller $\mathcal{S}$	Buyer $\mathcal{B}$
$k = \text{Random}()$ $C = E_k(\text{Data}_{\mathcal{S}})$ $h_1 = H(C)$ $h_2 = H(k)$ $\sigma = \text{Sign}_{Notary}(h_1 \| h_2 \| \mathcal{S}_{id})$		
$\text{Send}_{Seller}(k \| C \| \sigma)$		
	$\text{Send}_{Buyer}(\sigma \| C \| h_2)$	
		$\text{Verify}(\sigma)$ Check $H(C) = h_1$?
		$T(x) := Pay(Seller)$ if $H(x) = h_2$ $\text{Publish}(T)$
	$\text{Publish}_T(k)$	

There is an initial **setup phase**, during which the Notary $\mathcal{N}$ receives the Seller's associated information ($\text{Data}_{\mathcal{S}}$) and generates a certificate by performing the following steps:

1. Notary verifies the information $\text{Data}_{\mathcal{S}}$ received from the Seller $\mathcal{S}$. $\text{Data}_{\mathcal{S}}$ contains the actual Data as well as the Seller's identification $\mathcal{S}_{id}$ and meta-information about the type of data.
2. Notary generates a random key k and uses k to create an encrypted version of the data (C).
3. Notary generates commitments for the key and ciphertext (h_1 and h_2), by computing a secure hash H of the key and ciphertext.
4. Notary signs a string obtained by concatenating h_1, h_2 and the Seller's identification $\mathcal{S}_{id}$.
5. Finally the Notary sends to the Seller $\mathcal{S}$ a certificate containing the random key k, the encrypted data C and the signature σ.

During the **transaction phase** (when Buyer and Seller actually perform the Exchange of Digital Goods), they follow these steps:

1. Seller $\mathcal{S}$ sends to the Buyer the signature σ, the encrypted data C and h_2.
2. The Buyer can verify that the signature σ is correct.
3. The Buyer can check that it has a proper encryption of the data by verifying the opening of the commitment for C, that is by verifying whether $H(C)$ is equal to h_1.
4. If everything is correct, the Buyer publishes a transaction on the blockchain that will pay the Seller if the key is revealed. The payment is executed only if the Seller exhibits an x such that $H(x) = h_2$.
5. The Seller closes the transaction $\mathcal{T}$ by publishing k, effectively opening the commitment h_2.

Note that the Notary generates the Seller's certificate before (or independently from) the Data request by the Buyer. The Notary does not play any part in the protocol during the transaction phase.

After the transaction is completed, the Buyer $\mathcal{B}$ uses the encryption key k to gain access to the Seller's data. This mechanism, wherein certain content is maintained private until a particular event (the publication of k) occurs, is reminiscent of the family of cryptographic primitives called Secure Triggers [8].

3.2 Protocol SEDG2

In the Wibson Data Marketplace [1], the Notary $\mathcal{N}$ is paid for his services in respect to a Data Transaction $\mathcal{T}$ by receiving part of the tokens paid by the Buyer.

In this section we describe a variation of the SEDG protocol, that allows the Notary to be paid for his services simultaneously with the Data Seller $\mathcal{S}$. The solution is described in Table 2.

The difference with the protocol SEDG1 is to include the Notary identification $\mathcal{N}_{id}$ as part of the key commitment to guarantee the payment:

- During the **setup phase**, h_2 is computed as the hash of the random key k and the Notary identification $\mathcal{N}_{id}$.
- During the **transaction phase**, in order to close the transaction $\mathcal{T}$, the Seller $\mathcal{S}$ has to publish both the encryption key k and the identification $\mathcal{N}_{id}$.

Table 2: Secure Exchange of Digital Goods (protocol 2)

Notary $\mathcal{N}$	Seller $\mathcal{S}$	Buyer $\mathcal{B}$
$k = \text{Random}()$ $C = E_k(\text{Data}_{\mathcal{S}})$ $h_1 = H(C)$ $h_2 = H(k\|\|\mathcal{N}_{id})$ $\sigma = \text{Sign}_{Notary}(h_1\|\|h_2\|\|\mathcal{S}_{id})$		
$\text{Send}_{Seller}(k\|\|C\|\|\sigma)$		
	$\text{Send}_{Buyer}(\sigma\|\|C\|\|h_2)$	
		$\text{Verify}(\sigma)$ Check $H(C) = h_1$?
		$T(x, n) := \text{Pay}(Seller, n)$ if $H(x\|\|n) = h_2$ $\text{Publish}(T)$
	$\text{Publish}_T(k, \mathcal{N}_{id})$	

3.3 Protocol SEDG3

Finally, we tackle the problem of hiding the transaction $\mathcal{T}$ from the Notary $\mathcal{N}$. In this scenario, after receiving the certification from the Notary, the Data Seller $\mathcal{S}$ wants to be able to use the certification without the Notary learning that the certification is being used.

The following protocol uses a blinded commitment [9] to hide the transaction from the Notary. It uses a public generator G to create the commitments. A simple version based on the discrete-log assumption in described in Table 3.

The difference with the protocol SEDG1 is that it uses a finite field generator $G \in \mathbb{Z}_p$ for the commitment h_2:

- During the **setup phase**, h_2 is computed as a power of G to the random key k in the finite field $\mathbb{Z}_p$.
- During the **transaction phase**, the Buyer chooses an additional random number r and sends it to the Seller.
- To close the transaction $\mathcal{T}$, the Seller $\mathcal{S}$ has to publish the product $k \cdot r$, thus effectively hiding the original key k from the Notary. If the transaction is correct, when $\mathcal{S}$ publishes x, it holds that

$$G^x \equiv G^{(k \cdot r)} \equiv \left(G^k\right)^r \equiv {h_2}^r \mod p.$$

As a result of the protocol SEDG3, the Notary $\mathcal{N}$ will not be able to identify the published transaction $\mathcal{T}$ as belonging to Seller $\mathcal{S}$.

Table 3: Secure Exchange of Digital Goods (protocol 3)

Notary $\mathcal{N}$	Seller $\mathcal{S}$	Buyer $\mathcal{B}$
$k = \text{Random}()$ $C = E_k(\text{Data}_{\mathcal{S}})$ $h_1 = H(C)$ $h_2 \equiv G^k \mod p$ $\sigma = \text{Sign}_{Notary}(h_1 \| h_2 \| \mathcal{S}_{id})$		
$\text{Send}_{Seller}(k \| C \| \sigma)$		
	$\text{Send}_{Buyer}(\sigma \| C \| h_2)$	
		$\text{Verify}(\sigma)$ Check $H(C) = h_1$? $r = \text{Random}()$ $\text{Send}_{Seller}(r)$
		$T(x) := \text{Pay}(Seller)$ if $G^x \equiv {h_2}^r \mod p$
		$\text{Publish}(T)$
	$\text{Publish}_T(k \cdot r)$	

4 Conclusion

Here we proposed a solution to the problem of trading real-world private information using only cryptographic protocols and a public blockchain to guarantee the fairness of transactions. We described a protocol that we call "Secure Exchange of Digital Goods" (SEDG) between a Data Buyer $\mathcal{B}$ and a Data Seller $\mathcal{S}$. The protocol relies on a trusted third party $\mathcal{N}$, which also plays the role of Notary in the context of a decentralized Privacy-Preserving Data Marketplace (dPDM) such as the Wibson Marketplace [1].

This protocol converts the Exchange of Data into an atomic transaction where two things happen simultaneously:

- The Buyer $\mathcal{B}$ gets access to the Data, by learning the key that enables him to decrypt C (previously received encrypted data).
- The Seller $\mathcal{S}$ gets paid for his Data by revealing the key.

We also presented two variations of the base protocol:

- In SEDG2, the Notary gets paid for his services at the very same time that the Seller gets paid for his data.
- In SEDG3, the Data Transaction is hidden from the Notary $\mathcal{N}$ (who generated the certificates used during the transaction).

Acknowledgements

The authors thank Nicolás Ayala and Martin Manelli for their work on the implementation of these solutions.

References

[1] Matias Travizano, Carlos Sarraute, Gustavo Ajzenman, and Martin Minnoni. Wibson: A decentralized data marketplace. In *Proceedings of SIGBPS 2018 Workshop on Blockchain and Smart Contract*, 2018.

[2] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing* , pages 364–369. ACM, 1986.

[3] Matteo Campanelli, Rosario Gennaro, Steven Goldfeder, and Luca Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 229–243. ACM, 2017.

[4] Bitcoin Wiki. Zero knowledge contingent payment. `https://en.bitcoin.it/wiki/Zero_Knowledge_Contingent_Payment`, 2018. Accessed: 2018-10-08.

[5] Aureliano Calvo, Ariel Futoransky, and Carlos Sarraute. An oblivious password cracking server. In *Workshop de Seguridad Informática (WSegI) at 41st JAIIO, La Plata, Argentina*, 2012.

[6] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy (SP)*, pages 459–474. IEEE, 2014.

[7] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *Cryptol. ePrint Arch., Tech. Rep*, 46:2018, 2018.

[8] Ariel Futoransky, Emiliano Kargieman, Carlos Sarraute, and Ariel Waissbein. Foundations and applications for secure triggers. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):94–112, 2006.

[9] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual International Cryptology Conference*, pages 129–140. Springer, 1991.