# Disciplina: Data Disclosure Protocol

TeachMePlease, https://teachmeplease.com
Serokell, https://serokell.io

Version 0.1
December 11, 2017

**Abstract**

This document describes the two-party protocol of fair data trade used in the Disciplina blockchain platform.

## 1 Notation

Throughout this document we use the following notations:

| Notation | Description |
|---:|---|
| $\mathbf{A}$ | A party that takes part in the protocol |
| $\mathtt{H}(m)$ | Result of applying a collision-resistant hash-function $\mathtt{H}$ to a message $m$ |
| $\mathtt{mtree}(a)$ | Merkle tree of the data array $a$ |
| $\mathtt{root}(M)$ | Root element of the Merkle tree $M$ |
| $\mathtt{path}(e, M)$ | Path of the element $e$ in the Merkle tree $M$ |
| $k$ | Symmetric key |
| $pk_{\mathbf{A}}$, $sk_{\mathbf{A}}$ | Public and secret keys of $\mathbf{A}$ |
| $\mathtt{E}_k(m)$ | Symmetric encryption with the key $k$ |
| $\mathtt{E}_{\mathbf{A}}(m)$ | Asymmetric encryption with the key $pk_{\mathbf{A}}$* |
| $\mathtt{Sig}_{\mathbf{A}}(m)$ | Tuple $(\mathbf{A}, m, sig(sk_{\mathbf{A}}, H(m)))$, where $sig$ is a digital signature algorithm* |
| $\mathtt{sizeof}(m)$ | Size of $m$ in bytes |
| $\bigoplus$ | Binary string concatenation |

## 2 Preliminary steps

Suppose the seller $\mathbf{S}$ has some data $D$. Before the deal $\mathbf{S}$ ought to perform some preparation steps. $\mathbf{S}$ should:

1. Divide $D$ into $N$ chunks of size no more than 1 KiB:

$$D = \bigoplus_{i=1}^{N} d_i, \quad \mathtt{sizeof}(d_i) \leq 1 \text{ KiB} \tag{1}$$

2. Generate a symmetric key $k$

3. Encrypt each $d_i$ with $k$ and make an array of encrypted chunks:

$$D_{🔒} = \{\mathtt{E}_k(d_1),\ \mathtt{E}_k(d_2),\ \ldots,\ \mathtt{E}_k(d_N)\} \tag{2}$$

*The particular keys $pk_{\mathbf{A}}$ and $sk_{\mathbf{A}}$ belonging to the party $\mathbf{A}$ are generally deducible from the context

4. Compute a Merkle root of the encrypted chunks:

$$R = \texttt{root}(\texttt{mtree}(D_{\faLock})) \tag{3}$$

On this stage $R$ is a public knowledge, while $k, D_{\faLock}$ and all of the $d_i$ are kept hidden.

# 3 Protocol description

The protocol fairness is guaranteed by a contract on the public chain. The contract is able to hold money and is stateful: it is capable of storing a log $L$ with data. All the data that parties send to the contract are appended to $L$.

1. The buyer generates a new keypair $(pk_{\mathbf{B}},\ sk_{\mathbf{B}})$, creates the contract and sends the money to the contract address. Along with the money, **B** sends the public key $pk_{\mathbf{B}}$ of the newly generated keypair.
2. If **S** agrees to proceed, she also sends a predefined amount of money to the contract address.
3. **S** transfers the encrypted data chunks $D_{\faLock}$ to the buyer. **B** computes the Merkle root $R'$ of the received data $D_{\faLock}'$:
$$R' = \texttt{root}(\texttt{mtree}(D_{\faLock}')) \tag{4}$$
4. **B** makes a transaction with a receipt $\texttt{Sig}_{\mathbf{B}}(R')$ to the contract address.
5. **S** sends $\texttt{Sig}_{\mathbf{S}}(\{\texttt{E}_{\mathbf{B}}(k),\ R\})$ to the contract. The contract accepts it iff $R = R'$ (this implies that $D_{\faLock} = D_{\faLock}'$).
6. **B** decyphers and checks the received data. In case some data chunk $e_i \in D_{\faLock}$ is invalid, **B** sends a transaction with $\{sk_{\mathbf{B}},\ e_i,\ \texttt{path}(e_i,\ \texttt{mtree}(D_{\faLock}))\}$ to the contract. By doing so, **B** reveals the data chunk $d_i$ corresponding to the encrypted chunk $e_i$. She also shares proof that $e_i$ was indeed part of a Mekle tree with root $R$. The contract checks the validity of $d_i$ and decides whether **B** has rightfully accused **S** of cheating.

The on-chain communications of the parties (steps 2, 4, 5, 6) are bounded by a time frame $\tau$. In order for the transaction to be valid, the time $\Delta t$ passed since the previous on-chain step has to be less than or equal to $\tau$. In case $\Delta t > \tau$ the communication between the parties is considered over, and one of the protocol exit points (Sec. 4) is automatically triggered.

# 4 Protocol exit points

To decide on whether the communication is over, the protocol utilizes some timeout $\tau$ (e.g. 1 hour) which bounds the communications that should happen between **B** and **S**.

| $\Delta t > \tau$ **at step** | **Consequence** | **Interpretation** |
|---|---|---|
| 2 | **B**, **S** get their money back | **S** wasn't able to transfer the data to **B**. |
| 3 | | |
| 4 | | |
| 5 | **B**, **S** get their money back | **B** received the encrypted data, but **S** wasn't able to share the key $k$ for it |
| 6 | **S** gets all the money | **S** correctly shared data to **B** |
| Protocol finishes | Either **B** or **S** get all the money | The dispute situation. In case **B** proofs **S** cheated, **S** loses all her money. Otherwise, **B** loses her money for false accusation. |