Monetha private data exchange protocol
Academic proof

Sebastian Faust, Sasan Safai, Marius van der Wijden, Sebastian Stammler

1 The Monetha Protocol

Monetha protocol Π is executed by a *Data Requestor* R and a *Passport Owner* O. These parties do not interact directly with each other. Instead, they do it via a smart contract (called *Passport Contract*) in which the Passport Owner deposited some money. Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure encryption scheme (see, e.g,) and let H be a hash function. The protocol takes as input a security parameter 1^n, a public-key/secret-key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^n)$ generated by the key generation algorithm, a positive natural number $I \in \mathbb{N}^+$ and a set of data keys $K_i \xleftarrow{\$} \mathbb{K}$ with $i \in [1, ..., I]$ sampled uniformly at random from set $\mathbb{K}$ (of size exponential in n). For $i = 1, \ldots, I$ let $h_i := H(K_i)$. Let Δ denote the maximal blockchain response time. The sub-protocol for sending key $K := K_i$ (for some $i \in \{1, \ldots, n\}$) from Data Provider to Data Requestor is depicted on Figure 1 (on page 2). It can be executed multiple times for different keys K_i.

1.1 Privacy

For the sake of the analysis of the privacy property of the Monetha protocol, we model the hash function $H : \mathbb{K} \to \{0,1\}^n$ as a random oracle. That is, for every unique query to H, we get a truly random response. Note that this is only done to facilitate the analysis of the protocol (see [1]). In practice, the random oracle can be replaced by any cryptographic hash function.

For the privacy notion of the Monetha protocol Π, we assume an honest but curious adversary, meaning that the adversary does not interfere with the protocol but just observes the transcript. We want to achieve that the adversary cannot infer any information apart from the publicly available parameters by merely observing the transcript of the Monetha protocol. This can only be true in case both, the Data Requestor and the Passport Owner, behave honestly.

Let 1^n and I be as above and let $b \in \{0,1\}$. Consider a distinguisher $\mathcal{D}(1^n, I)$ that is an interactive machine that plays the following game against an oracle $\Omega(1^n, I, b)$.

1. The oracle chooses a set of data keys $K_i \xleftarrow{\$} \mathbb{K}$ with $i \in [1, ..., I]$ and a random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^n)$. For $i = 1, \ldots, I$ let $h_i := H(K_i)$. The oracle sends $(h_1, \ldots, h_I)$ and pk to the distinguisher.

Protocol $\Pi(1^n, (\mathsf{pk}, \mathsf{sk}), K, i)$

1. The Data Requestor R chooses uniformly a random a key $K' \xleftarrow{\$} \mathbb{K}$, computes the hash value $h' \leftarrow H(K')$ as well as the ciphertext $C \leftarrow \mathsf{Enc}(\mathsf{pk}, K')$ under the public key of the Passport Owner.
2. The Data Requestor sends (h', C, i) to the Passport Contract together with his money deposit.
3. The Passport Contract emits an event of the data request, upon which the Passport Owner reads the ciphertext C.
4. The Passport Owner O computes the key $K' \leftarrow \mathsf{Dec}(\mathsf{sk}, C)$ using his secret key sk. If $h' = H(K')$ then he computes $K'' = K \oplus K'$. Otherwise he halts.
5. The Passport Owner sends K'' to the Passport Contract as a reply to the event from step 3. If the Passport Owner does not send this reply within some time Δ then the Data Requestor requests his deposited coins to be given back to him.
6. The Data Requestor obtains K'' from the Passport Contract and he computes $\tilde{K} = K'' \oplus K'$. If $H(\tilde{K}) = h_i$ then he accepts $\tilde{K}$ as the valid key K. Otherwise he sends a complaint (together with K') to the Passport Contract. The contract performs the following steps:
   (a) If $H(K') \neq h'$ then the Passport Contract rejects the complaint.
   (b) If $H(K'' \oplus K') \neq h_i$ then the Data Requestor gets a refund of his deposited money and the money which was deposited by the Passport Owner.
   (c) Otherwise the Passport Owner gets both deposits.
7. If no complaint was issued by Data Requestor within the Δ, then Passport Owner claims back his deposit and receives the deposited money of the Data Requestor.

Define the *transcript of an execution of* Π *on input* $(1^n, I, (\mathsf{pk}, \mathsf{sk}), K, i)$ as

$$\mathit{Trans}_{\Pi}(1^n, (\mathsf{pk}, \mathsf{sk}), K, i) = (C, h', K'')$$

Figure 1: The sub-protocol of Monetha Protocol for sending key $K := K_i$ (for some $i \in \{1, \ldots, n\}$) from Data Provider to Data Requestor

2. The distinguisher can request an arbitrary number of independently sampled transcripts $\mathit{Trans}_\Pi(1^n, I, (\mathsf{pk}, \mathsf{sk}), K, i)$ (for any i). The oracle samples them for the distinguisher according to the algorithm from Fig. 1 (note that she can do it, since she knows sk).

3. Consider the following cases:

   - $b = 0$: then the oracle sends to the distinguisher $(X_1, \ldots, X_I)$, where the X_i's are sampled independently at random from the set $\mathbb{K}$.
   - $b = 1$: then the oracle sends to the distinguisher the tuple $(K_1, \ldots, K_I)$.

4. The distinguisher outputs a bit denoted

$$\mathrm{Out}(\mathcal{D}, 1^n, I, b).$$

Definition 1.1 (Transcript Privacy)**.** *The Monetha protocol* Π *is* transcript-private *if for every probabilistic polynomial-time distinguisher* $\mathcal{D}$ *we have*

$$|\mathbb{P}\left[\mathrm{Out}(\mathcal{D}, 1^n, I, 0) = 1\right] - \mathbb{P}\left[\mathrm{Out}(\mathcal{D}, 1^n, I, 1) = 1\right]| \leq \epsilon(n), \tag{1}$$

where ϵ *is some negligible function.*

Theorem 1 (Privacy)**.** *Assume* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is CPA-secure and* H *is modeled as a random oracle. Then the Monetha protocol* Π *is transcript-private.*

Proof. We first consider the case of one key only, i.e., we assume that for some fixed $\widehat{i}$ all transcripts are of a form $\mathit{Trans}_\Pi(1^n, (\mathsf{pk}, \mathsf{sk}), K, \widehat{i})$, and the only goal of the adversary is to distinguish $K_{\widehat{i}}$ from a random key. Intuitively this will suffice since all the transcripts $\mathit{Trans}_\Pi(1^n, I, (\mathsf{pk}, \mathsf{sk}), K, i)$ are independent (we show that this intuition is correct at the end on this proof).

Let $K = K_{\widehat{i}}$ and consider a modified game where the transcripts are of a form

$$\mathit{Trans}'_\Pi(1^n, (\mathsf{pk}, \mathsf{sk}), K, i) = (h', K'')$$

(i.e. they are as before, except that the ciphertext "C" is missing). We now argue that for every distinguisher $\mathcal{D}$ in the original game there exists a distinguisher $\mathcal{D}'$ in the modified game such that for every b we have

$$|\mathbb{P}\left[\mathrm{Out}(\mathcal{D}, 1^n, I, b) = 1\right] - \mathbb{P}\left[\mathrm{Out}(\mathcal{D}', 1^n, I, b) = 1\right]| \leq \mathsf{negl}(n) \tag{2}$$

This follows from the fact that $\mathcal{D}'$ can simulate transcripts $(\widehat{C}, h', K'')$ from (h', K''), by computing $\widehat{C}$ as $\widehat{C} := \mathsf{Enc}(\mathsf{pk}, Y)$ (for some fixed $X \in \mathbb{K}$). From the CPA-security of the encryption scheme it follows that the transcripts $(\widehat{C}, h', K'')$ have to be computationally indistinguishable from the original (C, h', K'')'s, and thus (2) holds.

Hence, it is enough to consider distinguisher $\mathcal{D}'$ that knows $H(K)$ and receives a number of transcripts of a form

$$\{(H(K'_a), K \oplus K'_a) \text{ where } K'_a \leftarrow \mathbb{K}\}_{\ell=1}^{m}$$

(for some natural number m) and whose goal is to distinguish K from a random key. Define an event Bad as follows:

$Bad :=$ "the distinguisher queried an oracle on K or one of the K'_a keys".

It is clear that as long as Bad did not happen the distribution of all the keys $(K, K'_1, \ldots, K_m)$ is completely uniform from the point of view of the distinguisher. Since the probability of guessing a uniformly random key is $1/|\mathbb{K}|$ and the running time of the distinguisher is polynomial, thus the overall probability of Bad is at most $p(n) \cdot |\mathbb{K}|$ (where p is some polynomial), which is negligible in n, since $|\mathbb{K}|$ is exponential in n. It is also clear that as long as Bad did not happen the distinguisher cannot distinguish K from a uniformly random $X \leftarrow \mathbb{K}$, and therefore

$$\mathbb{P}\left[\mathrm{Out}(\mathcal{D}', 1^n, I, 0) = 1 | \neg Bad\right] = \mathbb{P}\left[\mathrm{Out}(\mathcal{D}', 1^n, I, 1) = 1 | \neg Bad\right].$$

Combining the above with the fact that the probability of Bad is negligible we obtain that

$$\left|\mathbb{P}\left[\mathrm{Out}(\mathcal{D}', 1^n, I, 0) = 1\right] = \mathbb{P}\left[\mathrm{Out}(\mathcal{D}', 1^n, I, 1) = 1\right]\right| \leq \mathsf{negl}(n).$$

Further, by applying (2) we obtain

$$\left|\mathbb{P}\left[\mathrm{Out}(\mathcal{D}, 1^n, I, 0) = 1\right] = \mathbb{P}\left[\mathrm{Out}(\mathcal{D}, 1^n, I, 1) = 1\right]\right| \leq \mathsf{negl}(n).$$

What remains is to confirm our intuition that the "single key case" implies the case with multiple keys. Formally, this easily follows from a hybrid argument (for more on this technique see, e.g., [2]). For $j = 0, \ldots, I$ define the *ith hybrid* $\mathrm{Hyb}_j(1^n, I)$ by modifying the distinguishing game by substituting steps 3 and 4 by steps 3' and 4' (respectively) defined as

3'. The oracle sends to the distinguisher a string

$$(X_1, \ldots, X_j, K_{j+1}, \ldots, K_I),$$

where $X_i \leftarrow \mathbb{K}$ (for $i = 1, \ldots, j$).

4'. The distinguisher outputs a bit. This output will be denoted

$$\mathrm{Hyb}_j(1^n, I).$$

Clearly the variable $\mathrm{Out}(\mathcal{D}, 1^n, I, 0)$ is distributed identically to $\mathrm{Hyb}_I(1^n, I, b)$ and $\mathrm{Out}(\mathcal{D}, 1^n, I, 0)$ is distributed identically to $\mathrm{Hyb}_0(1^n, I, b)$. Hence, to prove that the left-hand-side of (1) is negligible, it suffices to show that for every $j \in \{0, \ldots, I-1\}$ we have that

$$\left|\mathbb{P}\left[\mathrm{Hyb}_j(1^n, I) = 1\right] - \mathbb{P}\left[\mathrm{Hyb}_{j+1}(1^n, I) = 1\right]\right| \tag{3}$$

is negligible. This, however, follows easily from the fact that the hybrids differ only on one key. Namely the jth key K_j is either equal to K_j or to a random X. Therefore it is identical to the "single key case" (as the values of all variables with indices other j can be internally simulated by the distinguisher). Hence: if (3) was non-negligible then this would contradict the "single key" security proven above. This finishes the proof. □

1.2 Fairness

Additionally to the transcript privacy property, we require the Monetha protocol to be fair, meaning that an honest party will never lose coins if it follows the protocol. More precisely, an honest Data Requestor can be sure that either he will get the requested information or he will get the deposited money back. Symmetrically, an honest Passport Provider can be sure that either he will get the required money for giving out some information or no information will be revealed. The disagreement between the parties is handled in steps (6a)—(6c) of the algorithm on Fig. 1. Since a dispute occurs already if only one data key is wrong, it suffices to analyze the dispute case for one key only.

Definition 1.2 (Fairness)**.** *Let Δ be blockchain's response time. The Monetha protocol is fair, if it fulfills the following two fairness properties:*

- Data Requestor Fairness: *If the Data Requestor is honest, he will either get the requested information or he will get the deposited coins back in time $O(\Delta)$. The deposit refund requires exactly one interaction with the blockchain.*
- Passport Owner Fairness: *If the Passport Owner is honest, he will get paid in time $O(\Delta)$ for revealing information.*

Fairness of our protocol holds under weaker assumptions than its privacy, namely it suffices to assume that H is collision-resistant (see, e.g., [2]). More formally, we have the following fact.

Theorem 2 (Fairness)**.** *Assume that H comes from a family of collision-resistant hash functions. Then the Monetha protocol is fair.*

Proof. We address separately the Data Requestor Fairness and Passport Owner Fairness. We start with discussing the Data Requestor Fairness. In case the Passport Owner is honest and sends the correct data key, there will be no dispute, since the Data Requestor received the correct key. Assume that the Passport Owner is malicious and sends a key $\tilde{K}$ to the Data Requestor that is not equal to $K'' \oplus K'$. The Data Requestor will detect that the key is not equal to K by comparing (in step 6) the publicly-known hash value h of the correct data key with $\tilde{K}$ (here we use the assumption that H comes from a family of collision-resistant hash functions). If the Data Requestor detects a wrong key, he complains to the Passport Contract, providing the key $\tilde{K}$. Since the Passport Contract can be seen as a trusted entity, it will reliably resolve the dispute by performing steps (6a)—(6c). Clearly case the Data Requestor gets a wrong data key, i.e. does not get the requested information, he will get his deposit back and he will additionally receive the deposit from the Passport Owner. The money will be sent to the Data Requestor after a certain time Δ, which is the time needed to execute the dispute in the smart contract. The Data Requestor needs exactly one interaction with the blockchain in case of a dispute, since he needs to call the function that initiates the dispute.

Passport Owner Fairness. An honest Passport Owner will always send the correct data key K to the Passport Contract, after the Data Requestor has made a deposit. If the Data Requestor is honest as well and accepts K, the Passport Contract will give the deposit from the Data Requestor and the deposit from the Passport Owner to the Passport Owner in time $O(\Delta)$.

If the Data Requestor is malicious and initiates the dispute, the Passport Owner will get the deposited money as well. First since in step (6a) the contract checks if $h(K') = h'$ therefore K' has to be equal to the key that O received as $\mathsf{Dec}(\mathsf{sk}, C)$. This is because in step 4 party O checks if $h' = H(\mathsf{Dec}(\mathsf{sk}, C))$ (note that here we use again the collision-resistance of H). Moreover (since O is honest) we know that for sure $H(K'' \oplus K) = h_i$. Hence the complaint will be rejected by the contract, and the money will be payed to the Passport Owner. It is easy to see that the Passport Owner gets this money in time $O(\Delta)$. □

References

[1] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *ACM CCS 93: 1st Conference on Computer and Communications Security*. Ed. by V. Ashby. Fairfax, Virginia, USA: ACM Press, 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[2] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. 2nd. Chapman & Hall/CRC, 2014. ISBN: 1466570261, 9781466570269.