Investigation of cryptographic primitives of atomic swaps performed on heterogeneous blockchains

Project work report
Beáta Éva Nagy
Advisors: János Tapolcai, Bence Ladóczki

1 Introduction to Blockchains

Blockchains are decentralized digital ledgers that are used to record transactions across computers in a peer-to-peer network securely and transparently. While first-generation blockchains merely function as distributed databases, second-generation blockchains introduce the concept of smart contracts, allowing the automatic execution of complex functions. Cryptocurrencies, such as Bitcoin and Ether are among the most well-known applications of blockchain technology.

In the world of cryptocurrencies, it is necessary to provide efficient methods for exchanging cryptocurrencies, i.e. for exchanging crypto tokens across different blockchain networks. Centralized cryptocurrency exchanges (CEXs) rely on a central authority to facilitate the exchanges. Decentralized exchanges (DEXs) improve decentralization by facilitating trades without storing the private keys of users [2]. However, complete decentralization can be executed through cross-chain atomic swaps [3], which do not need a third-party arbitrator at all. They perform the exchange in a single step, allowing parties that do not trust each other to trade tokens securely.

If both networks support smart contracts, the execution of atomic swaps can be facilitated easily [3]. However, it is a challenging task if one of the networks does not have scripting functionality. To find a solution in such cases, atomic swaps have been implemented for specific pairs of blockchains, each relying on their characteristics. However, with over 10,000 cryptocurrencies in existence [1], there is a need for the generalization of the process.

2 Atomic Swap Abstraction

The goal of the study is to examine a proposed abstraction of the atomic swap process, which is independent of the specific blockchains it is applied to. Figure 1 demonstrates the general steps of an atomic swap through an example: Alice and Bob exchange cryptocurrencies stored on different blockchains.

One of the most crucial parts of the process is the use of digital signatures: they ensure the authenticity and integrity of the transactions stored on the blockchains. To facilitate this, several widespread cryptographic signature schemes exist, based on public key cryptography. Both parties possess a private key that they do not share with anyone, and a publicly available key. In the context of digital signatures, the former

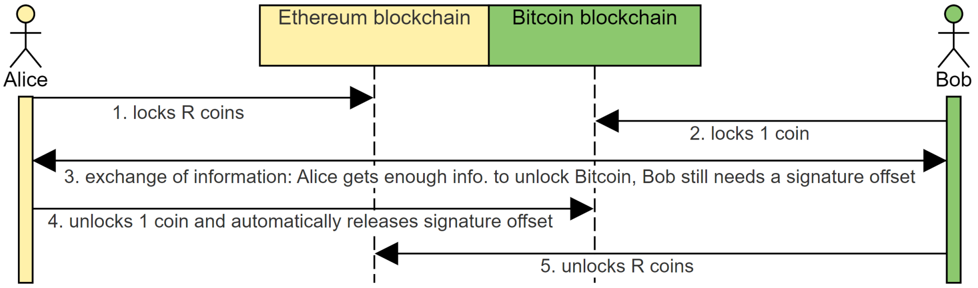
Ethereum blockchain
Bitcoin blockchain
Alice
Bob
1. locks R coins
2. locks 1 coin
3. exchange of information: Alice gets enough info. to unlock Bitcoin, Bob still needs a signature offset
4. unlocks 1 coin and automatically releases signature offset
5. unlocks R coins

Figure 1: *An example of an atomic swap process, where Alice sends R Ethers to Bob in exchange for 1 Bitcoin. The atomic nature of the process lies in the fact that the moment Alice unlocks the Bitcoin, Bob automatically gets the signature offset required to unlock the Ethers. Therefore, either both transactions happen or neither of them.*

is used for signing a message, while the latter is utilized by the other party, to verify the integrity of the message.

During the project work, I concentrated on the Elliptic Curve Digital Signature Algorithm (ECDSA) [7], and the Schnorr signature scheme [8], as they are efficient and suitable for cryptocurrency transactions. Using SageMath, I implemented the most important cryptographic primitives that form part of the atomic swap, concentrating on signatures.

3 Schnorr and ECDSA Based Implementation

Both signature schemes make use of the characteristics of *elliptic curves* over a finite field $\mathbb{F}_p$ (where p is a large prime), defined by the following general equation:

$$y^2 \equiv x^3 + ax + b \; (mod\; p),$$

where a and b are chosen elements of the finite field. Let q denote the order of the curve (the possible number of points on the elliptic curve, including the point at infinity). Let G be a given point of the elliptic curve, called the *generator* of the curve.

The set of points on the elliptic curve forms an abelian group with a commutative group law [4]. Given an arbitrary element of the finite field (denoted by z), if we take $z \times G$, the corresponding point (denoted by G') on the elliptic curve can be calculated in polynomial time. However, from G' and G the calculation of z is an NP-hard task, stated by the Elliptic Curve Discrete Logarithm Problem (ECDLP) [6]. Both signature schemes utilize this fact, as all secret values (such as the secret key) are elements of the given finite field, and the corresponding public values are calculated by multiplication with G.

In the case of Schnorr signatures, the signature is constructed in the following way:

$$s \equiv r + h \cdot x \;\; (mod \;\; q),$$

where s denotes the signature, r is a random element of the field, h is the hashed message, and x is the secret key. The other party obtains s, together with the public versions of r and x (computed by multiplying each element by G, denoted by R and X, respectively), and can compute h as well. Therefore, after computing $S = s \times G$, if

$$S = R \oplus h \times X,$$

then we can verify that s was indeed constructed from the private elements r and x, without actually revealing them.

ECDSA uses a slightly different congruence,

$$s \equiv (h + \mathcal{R} \cdot x) \cdot r^{-1} \;\; (mod \;\; q),$$

where $\mathcal{R}$ is constructed by taking the public version of r ($R = r \times G$), cutting out the leftmost bit and taking (mod q). In the verification process, after calculating $\mathcal{R}$ and h, if

$$s \times R = (h \times G) \oplus (\mathcal{R} \times X),$$

then the signature was indeed valid.

To improve security, valid signatures can be concealed using offsets, and an offset signature can be transmitted to the other party: $s' = s + t$, where t is a generated random offset. In this case, the primitives take $s + t$ instead of s. The verification of the public version of the offset, and computing the offset itself from the original and offset signatures can be executed similarly.

During the project, I implemented both Schnorr and ECDSA signature schemes, by writing functions that create the signatures, verify them, add offsets to the signatures, verify the offsets, obtain the offsets from the original and offset signatures, and de-offset signatures. These functions form the basis for the atomic swap process, and can be utilized when carrying out the entire process.

4 Conclusion, Future Work

Developing a general protocol that is capable of performing atomic exchanges between any pair of blockchains is a step towards improving the decentralization of the currency exchange process. Standardization can make atomic swaps more popular as an exchange method, and it can also eliminate the need for implementing atomic swaps for each pair of blockchains.

By researching the topic and implementing signature schemes, I significantly improved my understanding of the topic, and provided a base for future simulations of atomic swaps. In the future, I am planning to continue working with atomic swaps, possibly by implementing other popular signature schemes, such as the Boneh-Lynn-Shacham (BLS) [5]. Furthermore, another interesting direction would be the comparison of each signature scheme, by running a large number of simulations, in order to reveal the advantages and drawbacks of each.

References

[1] *CoinGecko*. `https://www.coingecko.com/`. (Visited on 12/17/2023).

[2] ETH Zürich. *Providing Liquidity in Uniswap V3*. 2022. URL: `https://pub.tik.ee.ethz.ch/students/2021-HS/BA-2021-21.pdf` (visited on 12/17/2023).

[3] Joël Gugger. *Bitcoin-Monero Cross-chain Atomic Swap*. Cryptology ePrint Archive, Paper 2020/1126. `https://eprint.iacr.org/2020/1126`. 2020. URL: `https://eprint.iacr.org/2020/1126`.

[4] Vorrapan Chandee et al. "Group Structures of Elliptic Curves Over Finite Fields". In: *International Mathematics Research Notices* 2014.19 (June 2013), pp. 5230–5248. ISSN: 1073-7928. DOI: `10.1093/imrn/rnt120`. URL: `http://dx.doi.org/10.1093/imrn/rnt120`.

[5] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Advances in Cryptology — ASIACRYPT 2001*. Ed. by Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 514–532. ISBN: 978-3-540-45682-7.

[6] Alfred Menezes. "Evaluation of security level of cryptography: the Elliptic Curve Discrete Logarithm Problem (ECDLP)". In: *University of Waterloo* 14 (2001).

[7] Neal Koblitz, Alfred Menezes, and Scott Vanstone. "The State of Elliptic Curve Cryptography". In: *Des. Codes Cryptography* 19 (Mar. 2000), pp. 173–193. DOI: `10.1023/A:1008354106356`.

[8] C. P. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology — CRYPTO' 89 Proceedings*. Ed. by Gilles Brassard. New York, NY: Springer New York, 1990, pp. 239–252. ISBN: 978-0-387-34805-6.