Atomic Bonded Cross-chain Debt

Mojtaba Tefagh
Department of Mathematical Sciences
Sharif University of Technology
Tehran, Iran
mtefagh@sharif.edu

Fatemeh Bagheri*
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
fateme.bagheri95@student.sharif.edu

Amirhossein Khajehpour*
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
amirhosseinkh@ce.sharif.edu

Melika Abdi*
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
melika.abdi@ee.sharif.edu

ABSTRACT

Inspired by the recent boom in *decentralized finance* (DeFi) and the unprecedented success of flash loan projects in this ecosystem, we introduce a decentralized debt derivative named *atomic bonded crosschain debt* (ABCD) to bridge the gap between the growth of lending protocols on Ethereum and other blockchains specifically Bitcoin. We think of ABCD as the alphabet of interoperability for DeFi and as a credit infrastructure which unlike the current protocols is not limited by requiring either smart contracts, over-collateralization, or instantaneous payback.

CCS CONCEPTS

• **Information systems** → E-commerce infrastructure; • **Networks** → Peer-to-peer protocols.

KEYWORDS

blockchain; decentralized finance; DeFi; unsecured loan; HTLC

ACM Reference Format:
Mojtaba Tefagh, Fatemeh Bagheri, Amirhossein Khajehpour, and Melika Abdi. 2020. Atomic Bonded Cross-chain Debt. In *Proceedings of 3rd International Conference on Blockchain Technology and Applications (ICBTA 2020).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Many financial instruments have been established and implemented in traditional fiat-based markets; among them: options, futures, loans, bonds, derivatives, etc. In the past decade, the concept of cryptocurrency has opened a new gate toward the next generation of economy and finance. This field is still open to new ideas and introduces lots of implementation challenges for DeFi.

By the invention of Ethereum smart contracts, so many decentralized financial applications were built which have resulted in the rapid growth of the ether market capital in general, and the total value locked in liquidity and lending protocols specifically [**? ? ? ? ?**]. This demonstrates the urgent need for the blockchain counterpart of well-known financial instruments, especially loans, options, and *decentralized exchanges* (DEX), and as a result, DeFi primitives are being demanded these days more than ever before [**? ? ? ?**].

In the present study, we tackle this challenge and design primitives for decentralized futures market applications which in addition to Ethereum blockchain, work on first-generation blockchains like bitcoin which do not support high-level Turing-complete scripting languages. Our bond issuance system and the corresponding procedures only require *hash time locked contracts* (HTLC) as their building block and they do not rely on any oracle or third party interference. To the best of our knowledge ABCD is the first protocol in DeFi which offers an atomic unsecured cross-chain bond service.

As the pioneer in decentralization, the pseudonymous Satoshi Nakamoto has devised a new path towards the peer-to-peer payment systems which are counted as a disruptive innovation today [**?**]. Ethereum as the next generation of decentralized computing services enables writing smart contracts on an electronic ledger [**?**]. Later on, by the advent of ever-increasing blockchains, one may need to exchange assets across different networks. Through utilizing atomic swaps, two parties on different blockchains make an atomic contract which transfers asset between them [**?**]. Up until now, several previous works have extended the usage of atomic swaps in different ways. Herlihy designed a model for analyzing atomic cross-chain swaps and suggested a protocol that not only removes incentives for any set of parties to deviate from the protocol, but also guarantees that no conforming party ends with the underwater outcome and showed that HTLCs are enough to achieved this [**?**]. Zamyatin et al. presented Xclaim which is a swap frame work based on the atomic

*These authors contributed equally to this work.

ICBTA 2020, December, 2020, Xi'an, China

ACM ISBN 978-1-4503-8896-2. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

swaps that is faster and considerably cheaper than normal atomic swaps [**?**]. The idea of atomic cross-chain transactions in Ethereum sidechains was developed in [**?**]. The conflict caused by the concurrent execution of smart contracts was addressed to make an all-or-nothing atomic cross-chain commitment protocol in [**?**]. Furthermore, Runchao et al. put a step forth by analyzing the fairness of atomic swaps and showed that the basic atomic swap is considerably more unfair compared to its equivalent contracts in the traditional market. Besides, they proposed two enhanced atomic swap protocols and justified their fairness [**?**]. Liu proposed an atomic swaption component which works only using low-level scripting tools [**?**]. Additionally, by utilizing his swaption component, offering fully decentralized futures contracts is no longer impossible [**?**]. Zie et al. extended the atomic cross-chain swap contracts to a new method that does not need HTLCs and everything is managed by different party's signatures [**?**].

The rest of this paper is organized as follows. First of all, in section 2 after defining the required terminology and presenting the other preliminaries of our work, we introduce the first practical *atomic bonded debt* (ABD) primitive. Later in section 3, we redesign our ABD model to build an intermediary ABD component, and by using it afterwards, we build the ABCD primitive.

2 ATOMIC BONDED DEBT

The idea behind ABD is inspired by flash swaps. In a flash swap, the loan is not paid to the borrower unless she pays it back immediately at the same block as borrowed [**?**]. In ABCD this is extended to several blocks i.e. the issuer can repurchase the bond in more than one block but it contains a secret and the bondholder's signature is required everywhere the capital is being utilized.

Unsecured bonds or debentures are not backed by some type of collateral. Since this bond is unsecured, the issuer does not have to deposit any margin, unlike the approach used in [**?**]. Assume Alice is the issuer and Bob is the purchaser of the bond. She needs to take the capital, exploit it in other contracts, and then repurchase the bond from Bob with some interest. She also needs Bob to deposit shortly after their contract has been started.

Fig. 1 shows the overall structure of ABD. In this figure, for each transaction, signatures, output amount, and locktimes are specified. Transactions' border colors show the party who broadcasts the transaction. In this type of bonds, all amounts have to be from the same coin. The cross-chain version of it is discussed later in the section 3.

Here, we employ HTLCs to make decisions. If the holder of a secret reveals it before the corresponding timelock deadline, the swap takes place. Otherwise, if she lets the locktime expire, the swap is reverted and all the locked values are given back to their initial owners. The secrets used in ABD are the *bond funding* key that the issuer uses to sell and the *exercise* key that the bondholder uses to exercise the bond. In this primitive, we use the Unix timestamp as the

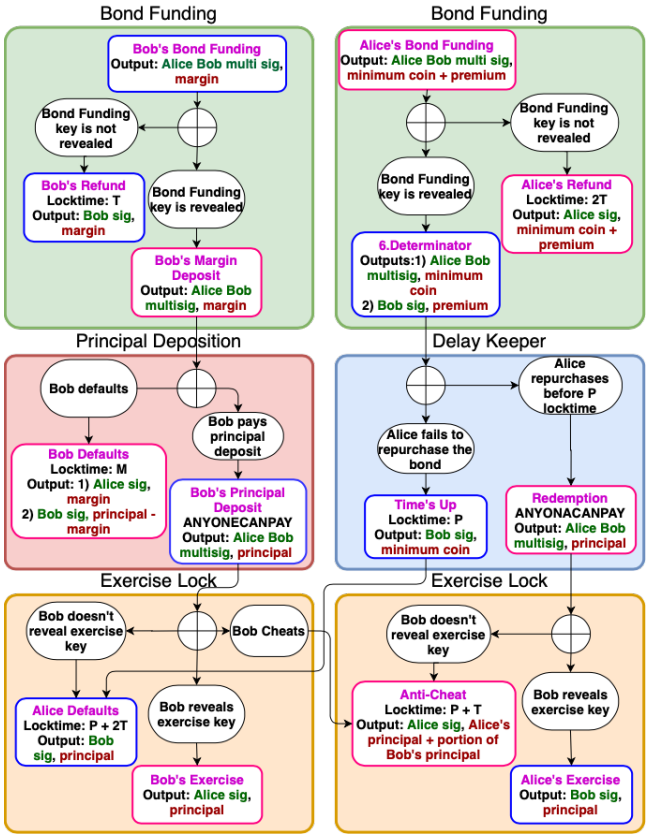

Figure 1: The ABD component. Each transaction is shown as a rectangle. On each transaction, signatures, output amount, and locktimes are specified. All outputs are in the same coin. Pink-bordered transactions are broadcast by Alice and blue-bordered ones by Bob. For locktimes, Unix timestamp is used. Upper transactions are broadcast earlier than the lower ones. If there is a line between two transactions, then the source transaction is considered to be an input of the destination transaction.

locktime parameter. We represent the minimum time needed for a mined transaction to be confirmed as T. In bitcoin, it is the time needed to have six subsequent blocks mined which is approximately one hour.

Next we explain in detail the process of exercising ABD which is basically made up of transactions shown in Fig. 1. First of all, all the transactions are signed and exchanged between the two parties except the bond funding transactions. In this phase, both parties make sure that there is no way the other party cheats on them, since either it is technically impossible or they get punished in case of cheating. This approach is similar to the procedure used by Poon et al. in the lightening payment channels [**?**]. After that, by broadcasting each of the transactions in the proper time, the process goes on. The procedure is divided into four different stages.

Depending on the application the bond is being used for, other stages can be appended to the procedure and transactions might need to be signed by more parties. However, here we explain the basic structure of the bond itself:

- **Bond Funding**: The funding transaction for Alice consists of
  - a premium,
  - and a very tiny amount for further usage (the minimum acceptable amount of output to be mined by the network miners, for example 546 Satoshis in bitcoin network at the time of writing).

  For Bob, the bond funding transaction only contains the margin. Alice has a relatively small amount of time to reveal the bond funding key to sell the bond. If she issues the bond, the premium goes to Bob and his margin goes to the Bob's margin deposit transaction.
- **Principal Deposition**: After selling the bond, each party has to deposit his principal within a specified time interval: Bob M locktime and Alice P ($P > M$). The *Bob's principal deposit* and the *redemption* transactions have sighash type of anyone-can-pay[1] since nobody knows all of their inputs in the first place. Bob can act either ways of:
  - If Bob defaults, then Alice takes his margin by broadcasting the *Bob defaults* transaction. Additionally, she will not fulfill the redepmtion transaction. Therefore, Bob can broadcast the *time's up* transaction, taking the minimum amount of coins which is too small to consider.
  - If Bob deposits the principal, the bond goes to the *delay keeper* stage.

  The *determinator* transaction and the minimum amount of coins are needed so that using this transaction we can force a deadline on Alice depositing her payback by utilizing the locktime on the time's up transaction. Also, in the case that Alice repurchases the bond, Bob can not claim that she did not broadcast the redemption transaction.
- **Delay Keeper**: At this stage, Alice has time before a P locktime expires ($P > M$) to fulfill her redemption transaction. If she does not deposit, Bob will broadcast the time's up transaction that prevents Alice from fulfilling her redemption transaction.
- **Exercise Lock**: The bond enters this stage when Bob deposits his principal. There are three possible scenarios in here:
  - In the last stages, Bob deposited his principal but Alice did not and the time's up transaction is broadcast. Now Bob does not reveal exercise key, and using the *Alice defaults* transaction he takes his principal back.
  - Bob has deposited his principal and Alice has fulfilled her redemption transaction but Bob avoids revealing the exercise key. Subsequently, Alice can broadcast the *anti-cheat* transaction which sends her principal and an amount of punishment from Bob's principal to her.
  - Both have deposited their principals. Bob reveals exercise key and they go to the next stage if there is any[2] and if not, each takes their coins and the procedure ends.

  Note that if Bob delays in broadcasting the time's up transaction, Alice may broadcast the redemption and anti-cheat transactions at the very last minute and cheat on Bob.

[1]The up-code ANYONECANPAY

3 ATOMIC BONDED CROSS-CHAIN DEBT

Since the Alice's time's up transaction in the ABD primitive needs to be shared in both Alice's and Bob's stages, this form of bond can not be used across different blockchains. To solve the problem of interoperability for the atomic bond, we have redesigned our model using check-sequence-verify[3]. In this section, we first build an intermediary component that has the same functionality as ABD. After that, we introduce ABCD that can be actually used between different blockchains. Using the check-sequence-verify, the minimum amount of coins for the determinator transaction and the delay keeper stage are removed from the component.

The intermediary ABD component is demonstrated in Fig. 2. To present this type of ABD we use block height as the locktime parameter, however the Unix timestamp can also be used. The procedure is discussed below:

- **Bond Funding**: Alice's funding includes premium and Bob's includes margin. The number C is the minimum number of mined blocks for a transaction to be confirmed.
- **Principal Deposition**: After issuing the bond, similar to the original ABD, each party has to deposit his principal within a specified time interval: Bob M locktime and Alice P ($P > M$). Bob's principal deposit transaction has sighash type of anyone-can-pay but Alice's redemption has "no-input" type since one of Bob's inputs is determined while none of Alice's inputs is clear at the time of creating the transaction. Bob can behave in one of the two following ways:
  - He defaults, then Alice takes his margin by broadcasting the Bob defaults transaction.
  - He deposits the principal, then the procedure goes to the next stage.
- **Exercise Lock**: If Bob deposits his principal before the locktime, Alice has time before a P locktime expires to repurchase her bond. As mentioned earlier, the redemption transaction has sighash type of no-input. The "checkseq 1" in the anti-cheat transaction, forces this transaction to have a minimum block distance of 1 from its parent i.e. the anti-cheat transaction can only

[2]In real-world applications there are usually more stages, since the bond is being used along with other contracts, otherwise it is useless giving ACoins and getting the same ACoins.

[3]The up-code CHECKSEQVERIFY

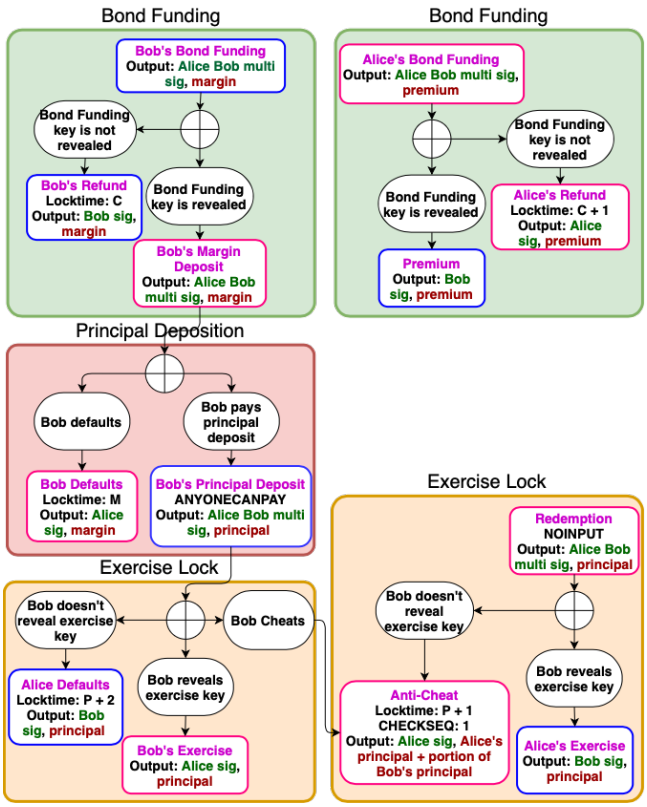

Figure 2: The ABD component using checkseq. On each transaction, signatures, output amount and locktimes are specified. All outputs are in the same coin. Pink-bordered transactions are broadcast by Alice and blue-bordered ones by Bob. For locktimes, block height is used. Upper transactions are broadcast earlier than the lower ones. If there is a line between two transactions, then the source transaction is considered to be an input of the destination transaction.

be mined in a block with a higher block height than the block which Alice's redemption transaction is in, hence Bob has enough time to reveal the exercise key. Note that the 1 block height difference may have to vary in different blockchains. For example in bitcoin, the minimum block height needed to elapse for the transactions to be confirmed is 6 blocks. Thus, we have to use "checkseq 6" in bitcoin. However, in this context, we assume the general case of 1 block.

After Bob's principal deposition, there are two possible scenarios:

- Alice succeeds to repurchase. Bob reveals the exercise key. The result is similar to the original ABD.
- Alice does not broadcast the redemption transaction hence Bob avoids exposing the exercise key and his principal is sent back to himself. To achieve this, Bob gives Alice $P + 2$ locktime to convince him to reveal the exercise key and if this deadline is passed, he will broadcast the Alice defaults transaction which gives him back his principal.
- Alice fulfills her redemption transaction but Bob does not reveal the exercise key. In this case, Alice can broadcast the anti-cheat transaction to get her principal and a portion of Bob's principal as the punishment. Note that, since the anti-cheat transaction has a locktime of $P + 1$ and a checkseq of 1 and the Alice's defaults transaction has a locktime of $P + 1$, Alice has to broadcast her redemption transaction before P. Otherwise, she would not have enough time to broadcast the anti-cheat transaction in the case of Bob cheating. In other words, if the Alice's redemption transaction is mined in a block of at least $P + 1$ height and Bob does not reveal exercise key, she can not get the anti-cheat transaction mined before block $P + 2$. Therefore, Bob can broadcast the Alice defaults in the $P + 1$th block and avoid giving her the capital.

The reason which prevents the intermediary ABD component to be used in a cross-chain environment is that the anti-cheat transaction has inputs from both Alice's and Bob's sides. To overcome this issue, before exploiting any bond contract, Bob, which is considered to be an exchange or somebody who has a reasonable amount of assets in different blockchains, can deposit some assets as a bond guarantee in any desired chain. This way, Bob can offer his customers a cross-chain bond service that accepts his payback on the other chain. This modification can be seen in Fig.3. In ABCD , we use Unix timestamp for locktimes and we consider the maximum time among all of involved blockchains since in different blockchains the number of blocks needed for confirmation is different.

The difference between the intermediary ABD component and the ABCD primitive is addition of the guarantee withdrawal transaction and its related funding transactios. Whether Alice defaults or the bond is successfully repurchased, Bob has to broadcast the *guarantee withdrawal* transaction. Additionally, in the case of not revealing the bond funding key, Bob can take back his guarantee. Other parts are the same in both procedures.

4 CONSLUSION

In this paper, we first introduced ABD which is a single-chain bond contract. Afterward, by extending its design, we derived ABCD to achieve the goal of providing an interoperable cross-chain bond. Collectively, we have employed the well-known atomic cross-chain swaps for building ABCD as a primitive for uncollateralized DeFi. Potential use cases include but are not limited to exploiting arbitrage opportunities between swaptions without owning any capital or any other similar use case of flash loans and flash swaps with two major improvements:

- Despite the similarities, instead of being a "flash" loan which must be repaid within a block, ABCD can span

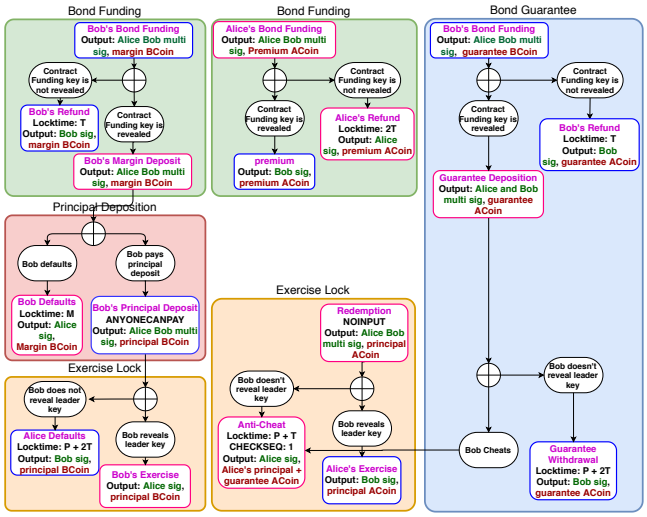

Figure 3: The ABCD across different chains. On each transaction, signatures, output amount and locktimes are specified. Bob's depositions are in BCoin and Alice's in ACoin. Pink-bordered transactions are broadcast by Alice and blue-bordered ones by Bob. For checkseq, block height is used. For locktimes, Unix timestamp is used. Upper transactions are broadcast earlier than the lower ones. If there is a line between two transaction, then the source transactions is considered to be an input of the destination transaction.

an arbitrary long period of time for the issuer to trade or invest with the capital before the bond reaches maturity. The significance of this feature is highlighted by noting that this is not possible even in conventional financial systems to have an unsecured debt without a credit system. More precisely, this property is only possible due to full transparency and traceability of cryptocurrencies.

- Our proposed bond primitive does not require a Turing-complete programming language and bitcoin scripting language is sufficient to implement our method which is completely based on HTLC. While most DeFi protocols rely heavily on smart contracts or third parties which make them susceptible to security issues, ABCD can be flexibly used on the wide variety of HTLC-compatible blockchains and in particular, supports bitcoin natively.

5 FUTURE WORK

As mentioned earlier, ABCD can be used along with other primitives in order to form more complex contracts. Depending on the application-specific domain, the current structure of ABCD might be either sufficient or not. In a future work, we aim to customize this structure to be fully compatible for integration in more complex systems.

On the other hand, there exist many variations of HTLC which in turn imply different categories of atomic swaps with different properties. The modular structure of our design enables similar variations on ABCD with the associated properties which can be explored in a future work.