A Protocol To Avoid Exchange Risk

vault_protocol@proton.me

November 2023

1 Introduction

In order to trade on centralized exchanges, a user needs to deposit collateral on the exchange. In case of exchange default, the collateral is usually lost which makes the interaction with centralized exchanges risky. Decentralized exchanges are in principle solving this problem, however, their adoption is still low compared to centralized exchanges. Furthermore, exchanges offering fiat on/off-ramps will very likely stay centralized (at least to a certain degree) for the foreseeable future. The proposed protocol tries to solve the centralized exchange risk problem.

2 Vault Protocol

2.1 Overview

In order to use the *Vault Protocol*, a user sends ERC20 tokens (e.g. ABC token) to the *Vault* contract. The *Vault* contract then issues two newly minted ERC20 tokens, cABC and iABC in equal amounts to the user. The user can redeem the original ABC token at any point by sending back the cABC and iABC tokens to the *Vault* contract where they are burned. The c stands for "collateral" and the i for "insurance".

Evidently, both tokens have value since together they can be used to redeem the original ABC token which is locked in the *Vault* contract. The following relation must hold:

$$value(ABC) = value(cABC) + value(iABC)$$

After depositing ABC tokens in the *Vault* contract and minting new cABC and iABC tokens, the user can then deposit the cABC tokens on some exchange and use it as collateral there. If the exchange does not accept cABC as collateral, the cABC token can be exchanged (e.g. via decentralized exchanges) for a token that the exchange accepts (effectively transferring the exchange risk to someone else willing to take it as the current cABC token price will reflect the perceived exchange risk and can be used for betting on the change of exchange risk or used as an exchange risk indicator which would be valuable in and of itself).

Upon default of the exchange, anyone can initiate a request to unlock the *Vault* contract. The exact procedure is explained in section 2.3 in more detail. If the *Vault* contract is unlocked, the original ABC token can be redeemed with iABC only, which means in the case of exchange default, the following condition holds:

$$value(ABC) = value(iABC)$$

2.2 Economic Model

Upon minting cABC and iABC tokens, a fee is deducted from the original ABC token balance and stored in the *Vault* contract. For example, if the user sends 100 ABC tokens to the *Vault* contract and the fee is set to 1%, the user would only receive 99 cABC and 99 iABC tokens which can later be redeemed for 99 original ABC tokens.

The accrued fees can be redeemed proportionally by all VAULT holders. The VAULT token is the governance token of the *Vault Protocol*. It is a simple ERC20 token with fixed supply and no minting or burning functionality.

In general, users will compare the utility of this protocol with buying insurance against exchange default directly. However, at the time of writing, buying insurance against exchange default is not straight forward or easily accessible. Centralized insurance offerings carry a risk that the payout will not be honored upon exchange default. Furthermore, insurance is usually expensive and premia need to be paid on a regular basis.

With the *Vault Protocol*, only a one time fee must be paid in order to be protected. There is, however, the caveat that the value of cABC is strictly less than ABC due to $value(ABC) = value(cABC) + value(iABC)$ and $value(cABC) > 0, value(iABC) > 0$ before exchange default. Therefore, less collateral value can be transferred to the exchange, limiting the maximal leverage that can be attained. However, iABC can be used as collateral for other (decentralized) services, minimizing the user's opportunity costs and increasing capital efficiency.

2.3 Vault Unlocking

A request to unlock the *Vault* contract can be initiated by any user. In order to initiate an unlock request, a certain amount of ABC tokens (that is dependent on the amount of outstanding iABC) needs to be deposited into the *Vault* contract. After the request has been issued, a voting period commences. During this period, all governance token holders can vote whether they believe the exchange specified by the *Vault* contract has really defaulted. Ideally, the voting is as anonymous and secret as possible for optimal convergence towards the truth (which is a Schelling point). However, even public voting should be sufficiently reliable given the outlined incentive structure. At the end of the voting period, the majority vote (the exact threshold can be defined) decides whether the *Vault* contract will be unlocked (and thus allowing to redeem ABC with iABC only). Two possible outcomes are possible:

The Vault Is Unlocked In this case, the initiating user is refunded the deposited ABC tokens. The governance tokens that voted against unlocking are

transferred to the users who voted in favor of unlocking.

The Vault Is Not Unlocked In this case, the ABC tokens of the initiating user are distributed among the governance token holders who voted against unlocking. The governance tokens who voted in favor of unlocking the Vault are transferred to the users who voted against unlocking.

2.4 Token Pricing

The token pricing will ultimately by decided by the forces of the free market subject to the constraints outlined in section 2.1. Intuitively, iABC should be priced higher than cABC due to it retaining the full value after an exchange default. The difference in price will therefore be proportional to the exchange default risk.

2.5 Usage

A *Vault* can be created permission-less for each (ERC20 Token, Exchange) pair. In principle, the "Exchange" can be replaced by any condition that can be resolved to true or false by an oracle. Apart from exchange defaults, a user can lose (or gain) cABC (or iABC) tokens in different ways like negative trading return or simply losing access to the private keys. A secondary market for cABC (or iABC) tokens will therefore establish itself. For example, if Alice is a very good trader and gains a lot of cABC tokens, she needs to buy iABC tokens somewhere in order to be able to redeem ABC from the *Vault*.

2.6 Attack Vectors

Any user A that loses cABC through bad trading or simply negligence will have an incentive to unlock the *Vault*. However, user A needs to assume that governance token holders are incentivized to keep the *Vault* shut since they will be rewarded with the deposited ABC tokens of the unlock initiator (likely user A) and also can expect to earn more fees from future users of the *Vault*. If user A holds the majority of governance tokens, the protocol becomes vulnerable.

If an exchange really defaults, it might look like governance token holders are in principle still incentivized to keep the *Vault* shut to earn the deposited ABC tokens of the unlock initiator, however, they would be destroying the value of their governance tokens since the credibility of the protocol would be shattered and market forces would adjust governance token value to 0.

Ideally, the amount of ABC tokens required to unlock a *Vault* should be high compared to the amount of outstanding iABC and cABC and low compared to the value of outstanding governance tokens. In practice this cannot be perfectly

achieved but increasing user adoption and tuning contract parameters will drive the protocol towards this goal.

2.7 Links

- https://github.com/the-vault-protocol/vault-protocol