Crin - Decentralized Insurance Protocol for token investment (DeIn)

Version 0.0.3 - updated: July, 28, 22
Tung le, tungle@decentralab.asia
Da Nang, July 2022

Crin protocol - the decentralized and automation protocol for crypto insurance. Crin allows everyone to issue insurance coverage to cover any ERC-20 token pair X/Y in a few seconds. Investors can recover the loss of investment when the token price goes down and the insurer can receive premiums as a reward for supplying insurance capital.

Đây là tài liệu được viết bằng tiếng Việt và có dùng xen lẫn rất nhiều thuật ngữ tiếng Anh. Vì vậy, nếu bạn cảm thấy khó chịu vui lòng đọc phiên bản tiếng Anh hoàn chỉnh ở đây. Chân thành cảm ơn!

Abstract

Đây là technical whitepaper giải thích cách mà Crin protocol v1 hoạt động. Chúng tôi sẽ giải thích cả về user flow cũng như logic và thuật toán ở trong smart contract. Chúng tôi sẽ tập trung giải quyết kiến trúc vận hành và cơ chế hoạt động nên sẽ chưa thực hiện giải quyết những vấn đề liên quan tới kinh doanh hay những lỗ hổng từ bên thứ 3. Dù Crin hỗ trợ multi-chain, nhưng trong tài liệu này, chúng tôi sẽ dùng mạng lưới Ethereum và DEX Uniswap như những ví dụ điển hình.

1 Introduction

Crin protocol cho phép bất kỳ ai sở hữu token X đều có thể tạo ra những gói Insurance Coverage để làm khoản bảo hiểm cho token Y khi giá X/Y biến động. Nhà đầu tư khi mua token Y sẽ được hưởng một khoản bảo hiểm để giảm thiểu rủi ro thua lỗ nếu giá X/Y đi xuống.

Thị trường crypto luôn biến động về giá rất mạnh và nhanh, nhất là khi DeFi ra đời với rất nhiều DEX cho phép bất kì ai cũng có thể tạo và giao dịch token. Chúng tôi muốn tạo ra cơ chế vừa bảo vệ những nhà đầu tư mới khi mua token, vừa tạo thêm nguồn thu nhập cho những người nắm giữ token.

Với Crin, chúng tôi mong muốn trở thành một mảnh ghép bền vững của DeFi. Bảo vệ những nhà đầu tư trước những dự án rug-pull, sự làm giá từ các cá mập và những lỗ hổng trong các smart contract cũng như protocols.

Crin được tạo ra hoàn toàn trên smart contract trên mạng ethereum, chúng tôi cũng sẽ hỗ trợ BNB chain và các chains khác trong tương lai. Vì vậy đây là giao thức phi tập trung và hoạt động tự động.

2 General

2.1 How it work

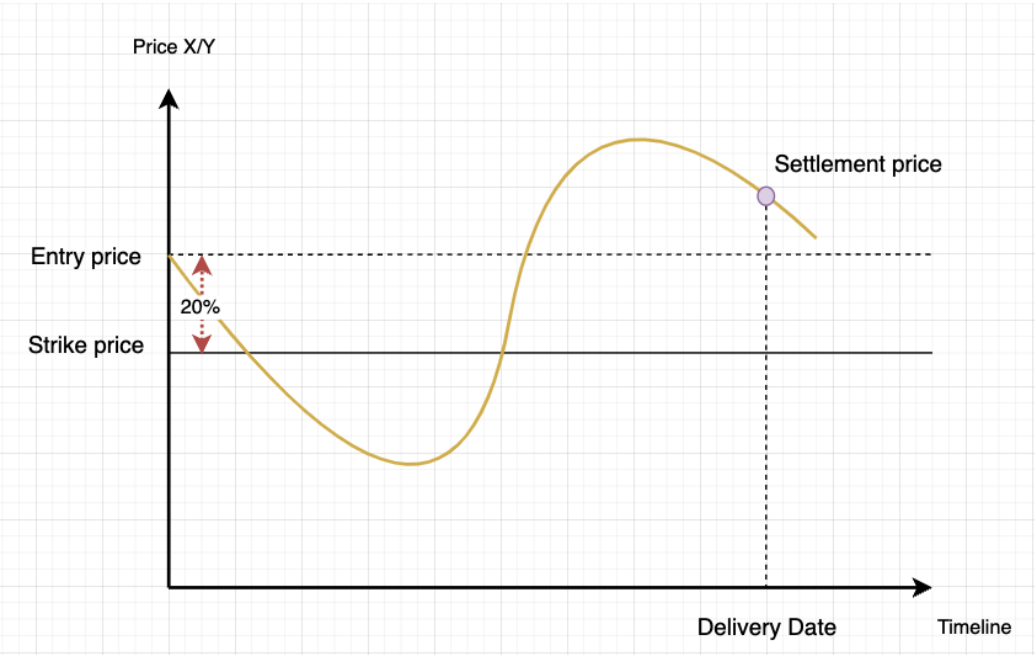
Ví dụ người dùng A tạo một gói bảo hiểm (insurance coverage) cho cặp token X/Y với khoản thua lỗ là 20%.

An investor B đầu tư token Y và muốn được bảo hiểm về giá trong một khoảng thời gian nhất định từ lúc mua. Khi đó:

- **Delivery Date**: Thời điểm kết thúc bảo hiểm
- **Entry Price:** Giá token X/Y tại thời điểm B đầu tư
- **Strike Price:** Giá token X/Y mà A sẽ bảo hiểm cho khoản đầu tư của B
- **Settlement Price:** A market price of X/Y in Deliver Date

Ngay khi đầu tư, B cần trả một khoản premium cho A vì dịch vụ bảo hiểm. Khi tới ngày kết thúc bảo hiểm. Sẽ có 2 trường hợp xảy ra:

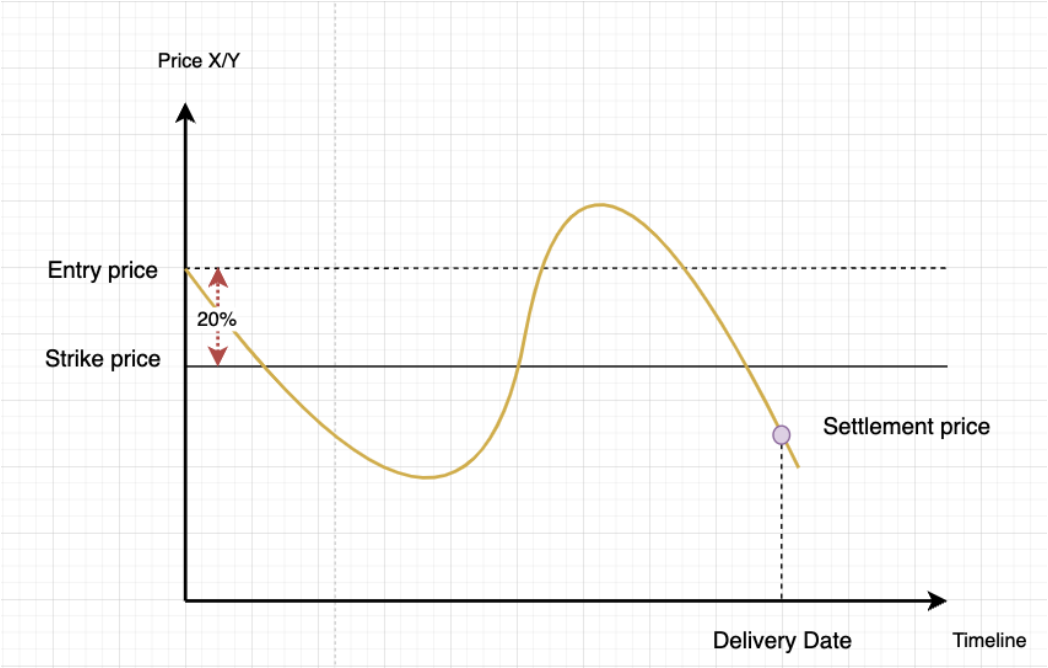
Trường hợp 1: Settlement Price lớn hơn Strike Price



A: _

B: Nhận về token Y đã đầu tư

Trường hợp 2: Settlement Price bé hơn hoặc bằng Strike Price



A: Trả khoản bù thua lỗ 20% cho B

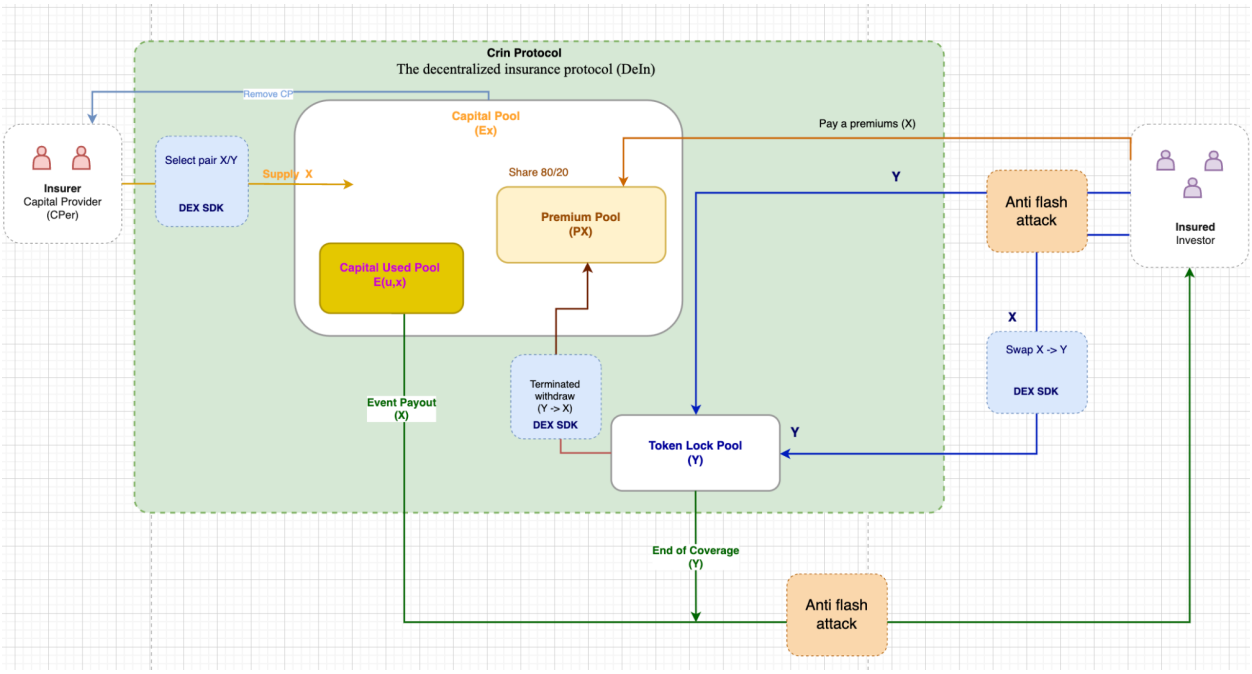
B: Nhận về token Y đã đầu tư cộng với khoản bù lỗ 20% của A.

—

Ví dụ thực tế:

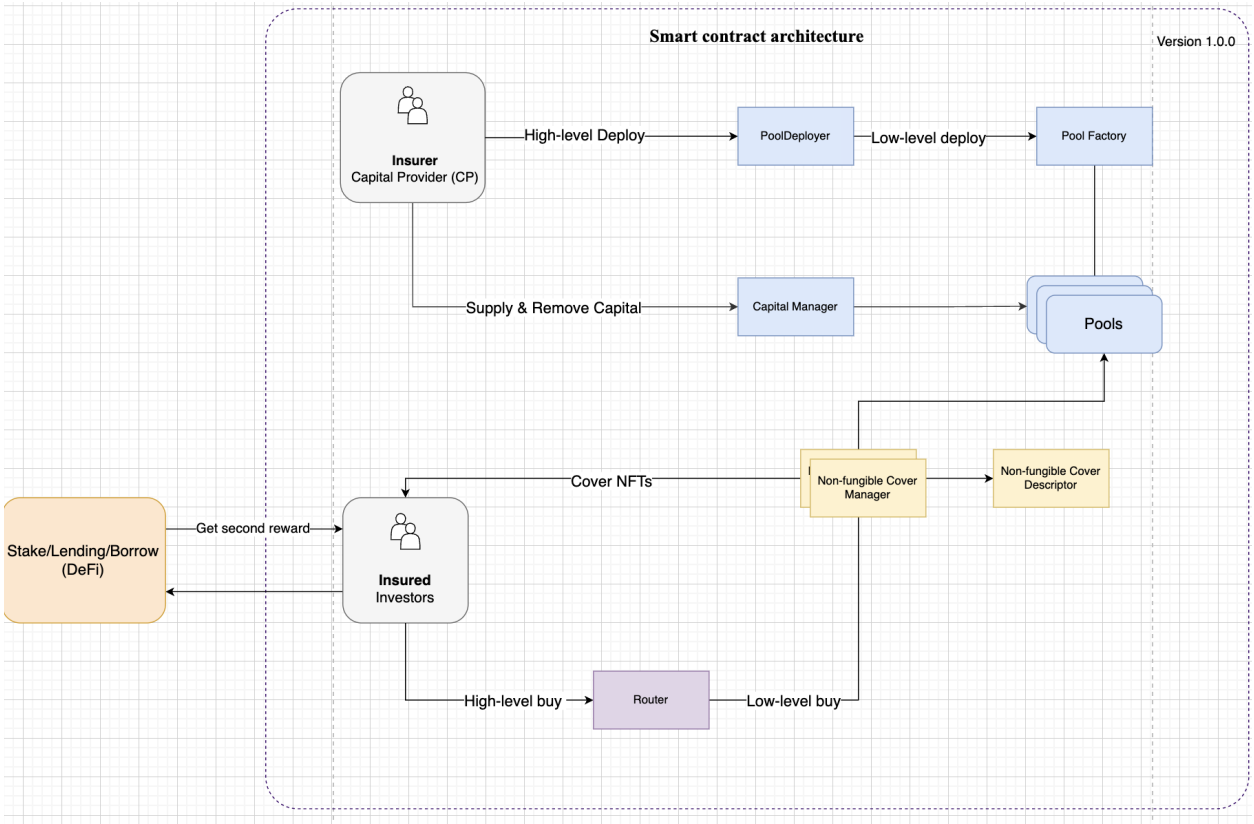
- A tạo một gói bảo hiểm cho cặp USDT/ETH với mức bảo hiểm thua lỗ là 20% và quỹ bảo hiểm là 1000 USDT
- B tham gia đầu tư 01 ETH ở mức giá USDT/ETH là 2000 trong vòng 3 tháng, khi đó B cần trả một khoản phí bảo hiểm (premium) là 100 USDT cho A.
- Sau 3 tháng, nếu:
    - Giá USDT/ETH lớn hơn 1600 thì B sẽ rút về 01 ETH đã đầu tư về ví cá nhân
    - Giá USDT/ETH nhỏ hơn hoặc bằng 1600 thì B sẽ rút về 01 ETH cộng với khoản bảo hiểm thua lỗ là 400 USDT

2.2 Protocol architecture



2.3 Smart contract architecture

Nhiều nguồn cảm hứng của Crin tới từ cách giải quyết vấn đề của Uniswap. Nên chúng tôi có học tập và xây dựng kiến trúc từ tài liệu của Uniswap v3[4]:



Full here:

2.4 Factory validate config

Các thông số được thiết lập ở factory smart contract để ràng buộc các dữ liệu đầu vào khi capital provider initialized new capital pool:

Type	Variable Name	Notaion	Default Value	Unit
uint24	periodMin	τ_{smin}	10	day
uint24	periodMax	τ_{smax}	365	day
uint256	dailyPremiumMin	ρ_{smin}	0.1	%
uint24	dailyPremiumMax	ρ_{smax}	1	%
uint256	dailyPremiumStep	ρ_{step}	0.1	

uint24	lossCoverMin	l_{smin}	10	%
uint24	lossCoverMax	l_{smax}	90	%
uint24	withdrawPenalty	$w_{penalty}$	1.5	%
uint24	withdrawShare	w_{share}	80	%

Table1. Global State in factory

3 Supply capital

3.0 Common variable

Everyone can initialized Capital Pool (CP) for pair X/Y if this pair was already on Liquidity Pool in Uniswap (swap-able) [2]. Với X/ Y pair ($f_{x,y}$) thì X được dùng làm đơn vị bảo hiểm cho biến động giá của X/Y ($P_{x,y}$) khi nhà đầu tư mua Y

Type	Variable Name	Notaion	Formula	Default Value
uint256	totalCapitalPool	Σ_x	$A = \{a_1, a_2, ...\} : a_i \in [0, \infty)$ $\Sigma_X = \sum_{i=1}^{\lvert A\rvert} a_i$	0
uint256	totalUsedCapital	Σ_X^u	$\Sigma_X^u = \sum_{i=1}^{\lvert A\rvert} a_i^u \wedge \Sigma_X^u \subset \Sigma_X$	0
uint256	premiumPool	Σ_ρ	$\Sigma_\rho = \sum_{i=1}^{\lvert B\rvert} \rho_{\beta i} : [0, \infty)$	0
uint24	tokenLockPool	Σ_Y	$B = \{\beta_1, \beta_2, ...\} : [0, \infty)$ $\Sigma_Y = \sum_{i=1}^{\lvert B\rvert} \beta_i : [0, \infty)$	0
uint24	safeAvailableRate	s_r		1 (%)

Table2. Smart contract common variable

A: Set of Capital of provider

Σ_x: Tổng khối lượng token X mà các capital providers đã supply vào pool.

Σ_X^u: Tổng khối lượng X đã dùng để đảm bảo cho insured

s_r: Để đảm bảo an toàn cho investor, tỷ lệ mức dư còn lại của CP Insured chỉ có thể đầu tư thêm nếu $\frac{\Sigma_x - \Sigma_X^u}{\Sigma_x} \geq s_r$

3.1 Initialized Capital Pool

When a capital provider initializes a new CP, Crin is allowed to optionally set insurance provisions in its CP. So with one token Y, there can be many CPs with different settings.

Type	Variable Name	Notaion	Domain
uint256	supplyCapital	c_i	$\{r \mid (0, \infty)\}$
uint256	dailyPremiumsRate	ρ_d	$\{r \mid [\rho_{smin}, \rho_{smax}] \ \wedge \ r \in B(\rho_{step})\}$
uint256	coverPeriodMin	τ_{min}	$\{r \mid [\tau_{smin} .. \tau_{max}]\}$
uint256	coverPeriodMax	τ_{max}	$\{r \mid [\tau_{min} .. \tau_{smax}]\}$
uint256	coverAmountMin	ι_{ymin}	$\{r \mid (0 , \iota_{ymax}]\}$
uint256	coverAmountMax	ι_{ymax}	$\{r \mid [\iota_{ymin}, S_y)\}$
uint24	lossCoverRate	l_{cr}	$\{r \mid [l_{smin} .. l_{smax}]\}$

Table3. Smart contract immutable variable

supplyCapital: Lượng token X để tạo CP ban đầu hoặc thêm mới

coverPeriod: Thiết lập khoảng thời gian [min,max] mà coverage có hiệu lực.

lossCoverRate: Khi % giá giảm tới mức này thì insured sẽ được nhận refund

amoutCoverRate: Tỷ lệ hoàn khoản thua lỗ khi insured ới event payout

$$\Sigma_x = c_i$$
$$A := \{c_i\}$$

3.2 Supply more capital

$$\Sigma_x := \Sigma_x + c$$

if

$$c_i \in A \Rightarrow a_i := a_i + c_i$$

Else

$$A := A \cup \{c_i\}$$

3.3 Remove capital

Sử dụng AFA để ngăn chặn những rủi ro có thể xảy ra.

To withdraw token X with c'_i amount from CP:

$$c'_i \in A$$
$$\Updownarrow$$
$$a_i^u = 0 \ \vee s_r \leq 1 - \frac{c'_i}{a_i - a_i^u}$$
$$\Updownarrow$$
$$a_i := a_i - c'_i$$

4 The insured flow

Type	Variable Name	Notaion	Domain
uint256	realtimePrice	$p_{xy}(\Delta_t)$	$(0, \infty)$
uint256	coverPeriod	τ	$\{r \mid [\tau_{min} .. \tau_{max}]\}$
uint256	amountToken1	$Y(\beta)$	$\{r \mid [\iota_{ymin}, \iota_{ymax}]\}$
uint256	entryPrice	p_e	$(0, \infty)$
uint256	amountToken0	$X(\beta)$	$[0, \infty)$

uint256	usedAmountToken0	$X^u(\beta)$	$(0, \infty)$
uint256	startTime	t_s	$(0, \infty)$
uint256	endTime	t_e	(t_s, ∞)
map	insurers	A^u	$A^u = \{a_1^u(\beta),\ a_2^u(\beta),\ \ldots\}$

Table4. Variable for the insured (NFTs)

$p_{xy}(\Delta_t)$: Get real-time price of X/Y pair in t through Uniswap SDK [2] and validate with AFA mechanism

τ: Khoảng thời gian muốn được bảo hiểm

$Y(\beta)$: Token Y với số lượng β muốn được bảo hiểm

$X(\beta)$: Số lượng token X tương ứng $Y(\beta)$ tại giá $p_{xy}(t)$

$X^u(\beta)$: Số token X được bảo hiểm nếu investor reach event payout

A^u: Danh sách các capital providers tham gia ký quỹ cho bảo hiểm β này

4.1 Insured invest

4.1.0 Condition

Sử dụng cơ chế AFA để tính đúng giá $p_{xy}(\Delta_t)$ tại thời điểm $t = t_{now}$. Ngăn chặn hành vi làm giá để gian lận bảo hiểm.

Khi investor muốn đầu tư một lượng token Y là $Y(\beta)$ thì điều kiện tiên quyết để có thể mua gói đầu tư có bảo hiểm là zoom bảo hiểm vẫn còn khả dụng ở trên mức s_r. Sao đó tiến hành lấy giá token của X/Y là $p_{xy}(\Delta_t)$ từ Uniswap SDK, thời gian hiện tại rồi gán vào các biến.

$$\frac{\Sigma_x - \Sigma_X^u - Y(\beta)}{\Sigma_x} \cdot 100 \geq s_r$$

$$\Downarrow$$

$$p_e = p_{xy}(\Delta_t)$$

$$t_s = t_{now} \wedge t_e = t_s + \tau$$

$$X(\beta) = p_{xy}(\Delta_t) \cdot Y(\beta)$$

Crin cho phép investor được đầu tư token Y với khối lượng a_y ở cả 2 trường hợp. Investor đầu tư Y thông qua token X hoặc bằng chính token Y đã sở hữu từ trước đó.

4.1.1 Invest $Y(\beta)$ with X

Khi thanh toán bằng token X, thực hiện tính toán tổng lượng token X là $\Delta_{pay(x)}$ cần để thành toán.

$$\rho'_x = X(\beta) \cdot \frac{(\rho_d * \tau)}{100}$$

$$\Delta_{pay(x)} = X(\beta) + \rho'_x$$

Sau khi investor thanh toán thành công thì swap $X(\beta)$ qua $Y(\beta)$ thông qua Uniswap tại mức giá tại thời điểm (t).

$$Y(\beta) = Uniswap(X(\beta))$$

4.1.2 Invest $Y(\beta)$ with Y

Khi thanh toán bằng token Y, thực hiện tính toán tổng lượng token Y là $\Delta_{pay(y)}$ cần để thanh toán.

$$\rho_y = Y(\beta) \cdot \frac{(\rho_d * \tau)}{100}$$

$$\Delta_{pay(y)} = Y(\beta) + \rho_y$$

Vì Premium Pool của Crin chỉ tính bằng token X, nên cần swap phí từ $\rho_y \rightarrow \rho'_x$.

$$\rho'_x = Uniswap(\rho_y)$$

4.1.3 Update database

Từ (4.1.1) hoặc (4.1.2), tiếp tục tính khối lượng token X dùng để bảo hiểm cho khoản β là $X^u(\beta)$

$$X^u(\beta) = \frac{l_{cr} \cdot X(\beta)}{100}$$

Khoản bảo hiểm $X^u(\beta)$ sẽ được chia đều theo tỷ lệ k_i tương ứng với mức mà mỗi a_i chiếm trong tổng CP Σ_x nên.

$$k_i = \frac{a_i}{\Sigma_x}$$

$$X^u(\beta) = \sum_{i=1}^{|A|} a_i^u(\beta)$$

$$\Rightarrow a_i^u(\beta) = k_i \cdot \frac{l_{cr} \cdot X(\beta)}{100}$$

Với mỗi a_i là vị trí của a trong A.

$$A^u := A^u \cup a_i^u(\beta)$$

$$a_i^u(\beta) = X^u(\beta) \cdot k_i$$

$$a_i := a_i + \rho'_x \cdot k_i \wedge a_i^u := a_i^u + a_i^u(\beta)$$

$$B := B \cup \{\beta\}$$

---

t_{now}: current time in second.

$\Delta_{pay(x/y)}$: Total token X or Y user need to pay

* System will automation swap a_x into a_y with price $p_{xy}(t)$ through Uniswap SDK [2]

4.2 End of coverage

Sử dụng cơ chế AFA để tính đúng giá $p_{xy}(\Delta_t)$ tại thời điểm $t = t_e = t_{now}$. Để dù insured có claim sau khi kết thúc nhiều ngày vẫn đảm bảo tính đúng đắn của bảo hiểm.

$$t_e \leq t_{now}$$
$$\Downarrow$$

Xác định tại thời điểm t_e kết thúc bảo hiểm. Mức thua lỗ $\frac{p_{xy}(\Delta_t) - p_e}{p_e}$ của Insured đã vượt qua mức l_{cr} hay không. Nếu có thì thực hiện tất toán bảo hiểm ở mức $X^u(\beta)$ và nhận về $Y(\beta)$ token Y. Còn nếu không thì Insured sẽ nhận về mình $Y(\beta)$. Sau đó cập nhật lại giá trị ở các pool.

$$E = (-1) \cdot l_{cr} \geq \frac{p_{xy}(\Delta_t) - p_e}{p_e} \cdot 100$$

$\top(E)$:

$$\Delta c_{yx} = Y(\beta) + X^u(\beta)$$

$$B := B \setminus \{\beta\}$$

$\perp(E)$:

$$\Delta c_y = Y(\beta)$$

Với mỗi i là index trong A^u.

$$a_i^u := a_i^u - a_i^u(\beta) \wedge B := B \setminus \{\beta\}$$

$\Delta c_{yx}/\Delta c_y$: Amount of token X and Y the insured can claim into the wallet

4.3 Terminated withdraw

Insured được phép cancel gói bảo hiểm hiện tại khi chưa tới thời điểm hết hạn t_e. Nhưng sẽ phải chịu một mức phí phạt $\rho_{penalty}$. Mức khí này sẽ được đưa vào Premium Pool (ρ_x). Còn Insured sẽ nhận về token Y với một lượng $\Delta_{withdraw}$ sau khi đã trừ đi phí phạt. Sau đó cập nhật lại giá trị ở các pool.

$$t_e > t_{now}$$
$$\Downarrow$$

$$\rho_{penalty} = w_{penalty} \cdot Y(\beta)$$

$$\Delta_{withdraw} = Y(\beta) - \rho_{penalty}$$

$$\rho'_x = Uniswap(\rho_{penalty})$$

Với mỗi i là index trong A^u.

$$a_i := a_i + \rho'_x \cdot \frac{a_i^u(\beta)}{X^u(\beta)}$$
$$a_i^u := a_i^u - a_i^u(\beta) \wedge B := B \setminus \{\beta\}$$

7 Anti flash attack (AFA) *(Mr Hieplq)*

Vấn đề: người tham gia bảo hiểm (tạm gọi là A) có khả năng lợi dụng tấn công flash-loan và front-run để gian lận bảo hiểm, đặc biệt là với những token có thanh khoản thấp.

- Trường hợp 1: Khi tham gia bảo hiểm, A có thể thực hiện front-run một lệnh mua với volume lớn khiến giá tăng cao hơn so với giá trị thực trong thời gian ngắn, sau đó mới tham gia bảo hiểm. Điều này sẽ khiến cho giá token lúc tham gia bảo hiểm của A lớn hơn giá trị hiện tại dẫn đến A có đủ điều kiện nhận bồi thường bảo hiểm nếu giá đi ngang hoặc ít biến động.
- Trường hợp 2: Lúc kết thúc thời gian bảo hiểm, A có thể thực hiện front-run một lệnh mua với volume lớn khiến giá giảm sâu hơn so với giá trị thực trong thời gian ngắn, sau đó mới thực hiện rút bảo hiểm. Điều này khiến giá token giảm sâu chạm mức kích hoạt bảo hiểm.

Giải pháp: để có thể giải quyết vấn đề này, chúng tôi đề xuất cơ chế mang tên anti flash attack (AFA), đây là cơ chế ngăn chặn người tham gia bảo hiểm cố tình sử dụng kỹ thuật flash-loan và front-run để thao túng giá cả, từ đó tạo ra điều kiện giá thuận lợi cho mình để gian lận bảo hiểm. AFA sẽ tính toán giá trị trung bình của token trong khoảng thời gian T1 (trước lúc tham gia bảo hiểm) và T2 (sau khi tham gia bảo hiểm), sau đó tiến hành so sánh với giá của token ở thời điểm tham gia (P) để xác định xem P là mức giá hợp lý (không chênh lệch quá nhiều so với giá trung bình - kiểm soát bằng hệ số alpha) hay không, từ đó quyết định giá tham gia bảo hiểm (hoặc huỷ yêu cầu tham gia - cái này

ae thảo luận). Đối với việc tất toán bảo hiểm cũng tương tự, AFA sẽ tính toán giá trung bình để thực hiện tất toán bảo hiểm dựa vào trung bình giá trong khoảng thời gian T. Việc giao tiếp với AFA và smart-contract đều thông qua oracle.

$$p_e = p_{xy}(\Delta_t) \text{ with } (\Delta_t \text{ in oracle})$$

8 User interface algorithm

https://drive.google.com/file/d/1ySN21w4DUEOXgSGYE-sCSn0q53Th-Den/view?usp=sharing

9 Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. This paper reflects current opinions of the authors and is not made on behalf of Decentralab or its affiliates and does not necessarily reflect the opinions of Decentralab, its affiliates or individuals associated with Decentralab. The opinions reflected herein are subject to change without being updated.

References

[1] https://docs.uniswap.org/protocol/V2/concepts/advanced-topics/pricing
[2] https://docs.uniswap.org/sdk/2.0.0/guides/trading
[3] Example calculated: https://docs.google.com/spreadsheets/d/1O2sKHtOMMb-P1jkSWhOlLs1_5j5_Pz1W3rM1kedbEC0/edit?usp=sharing
[4] Uniswap v3: https://uniswap.org/blog/uniswap-v3
[5] Crin protocol github: https://github.com/Crin-Protocol