Zano Tokenization Platform

Pavel Nikienkov[1], sowle[2] Andrey N. Sabelnikov[3],

[1] pavel@zano.org
[2] val@zano.org [3] andre@zano.org

October 27, 2022

1 Introduction

Tokenization has played a huge role in the development of the modern cryptocurrency ecosystem, and it is difficult to overestimate the practical opportunities it provides. The ideas that were once born in Ethereum have gone a long way of evolution and have formed a whole ecosystem of cryptocurrency products, largely based on tokenomics. The first rudimentary ideas of tokens in the blockchain industry began to sound back in 2011, back then it were called "colored coins"([1]), since the idea was to mark certain coins with some attribute (color). At that time, crypto community enthusiasts were looking for the possibility of representing objects from the real world through crypto-assets (tokens).

With the Ethereum arrival, the industry has reached a new level. Built on the principle of the Turing-complete virtual machine, the Ethereum network provided the opportunity not only to create new tokens of various configurations but also made it possible to create more complex financial instruments based on these tokens, stable coins, liquidity pools, collateral borrow protocols, NFT exchanges, etc., everything that is now known under the term "DeFi".

In parallel with the development of the ecosystem of tokens, and independently of this, privacy-oriented projects have developed. Transactions in such projects do not reveal either the addresses of the senders or recipients of this transaction or the amount of money transferred and make it difficult to link the chain of transactions to each other. All of the above properties are important for privacy, but make it impossible to build DeFi tools within the ecosystem of private projects, on the other hand, there is definitely a demand for privacy among DeFi users, and currently existing products (such as tornado.cash) can solve this problem due to the nature of EVM (Ethereum Virtual Machine) networks.

We at Zano have came up with solution a solution that allows users to issue tokens that work inside our blockchain, which have the same properties as transactions with a native token—namely hidden amounts and hidden addresses, as well as auditable wallets. Beside building our own ecosystem, we provide integration with DeFi ecosystem by creating mechanisms that will allow bridging tokens created in Zano platform to other EVM-like blockchains, like Ethereum.

2 Implementation

Normally transactions in CryptoNote-family blockchain use "stealth address" [4] to hide address from transactions and use mixins [2], to unlink transactions chains, so every input in a transaction contains set of decoys with only one real input reference. Ring signature proves that transaction creator owns one of those inputs, but it's assumed as not possible to determine which one from those decoys are real one.

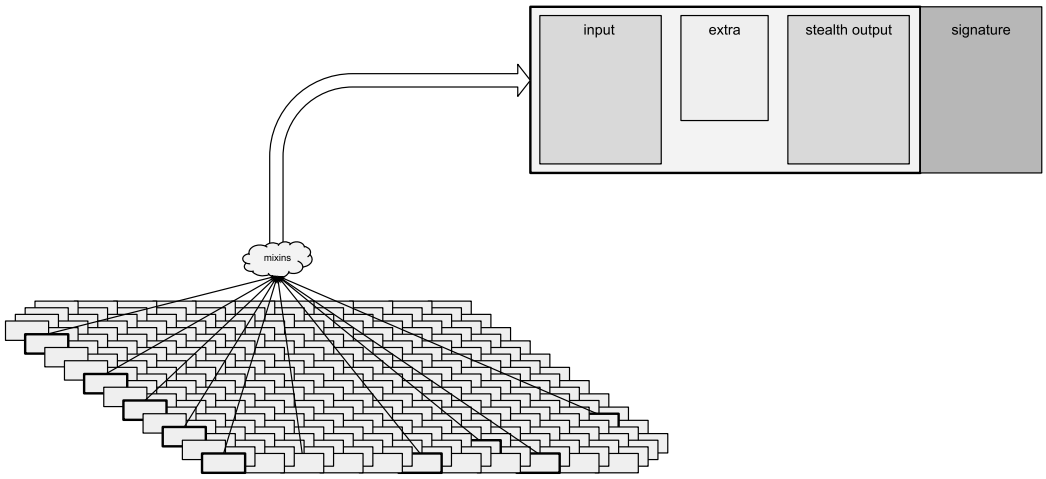

Fig. 1. A classic CryptoNote Transaction

Tokens are introduced to Zano along with the Zarcanum hard fork [3], which moves the Zano network to a completely anonymous PoS.

A new token issued with special transaction (Fig. 1), which define a new entity (let's call it "token descriptor"), all parameters of the future token will be stored in this descriptor (such as: the maximum emission, current number of tokens in circulation, a public key of an update authority, the as well as a unique token id).

2.1 From Aliases to Tokens

From the very creation of Zano, there was such a thing as **alias**, this is a mechanism that allowed the creation of named entities in Zano's private transaction space, roughly speaking an analogue of the domain name system, which associated a text name with a public address, and also determined the mechanisms for transferring ownership of this object. By scaling this solution, we form a mechanism for the declaration of token identity, its properties and attributes. Thus, in addition to a unique identifier, a token can be identified by a unique name that was registered along with the creation of the token. The previously formed mechanisms for transferring ownership of aliases, as well as its editing, will allow user to maintain the parameters of tokens.

2.2 Mixing and concealing

In Zano Tokenization Platform, all tokens mimic regular native transactions, that means that after issuing a new token, all its outputs go into stealth addresses, which would be used later by mixing it with other tokens outputs and native transactions. That creates a very unique model of network: transactions operate with diverse assets, but from a third-party observer it won't be possible to make a difference if a given transaction operates native coin or some synthesized token. Moreover, with a mixins will be included outputs of any tokens and native coins, and it is cryptographically proven, that resulting amount of the tokens are same type and same summary amount, without revealing its type and its amount.

In the Fig. 2 we projected simplified visualization of how tokenized transactions would be structured in the blockchain with 6 tokens present. That transaction does transfer a native coins, coins of "Token 2" and

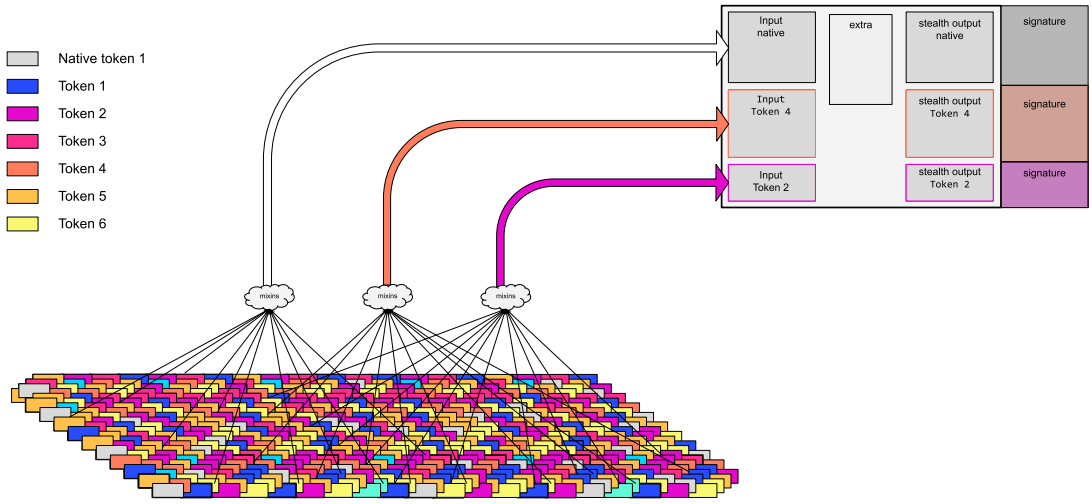

Fig. 2. Zano Tokenized Transaction

coins of "token 4". As it is shown, for selecting decoys for every input it uses random outputs, regardless of its type.

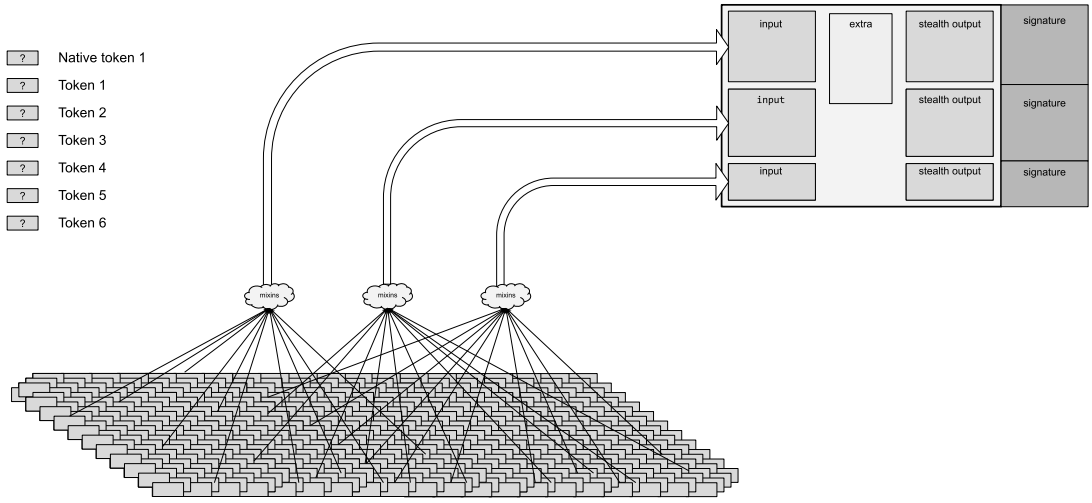

Fig. 3. Zano Tokenized Transaction as see by a third party observer

To illustrate the indistinguishable nature of tokenized transaction described above, we show at Fig. 3 the representation of this transaction as it is seen by a third party observer, who is trying to make open blockchain analysis.

All information about token identification is blinded with commitments, the same way as it used to hide amounts in transactions. Within a single transaction, it will be possible to combine the use of native coins with synthesized tokens, with an appropriate separation of outputs. At the same time, transaction outputs that are tokens, cannot be used for staking, otherwise the financial model would be broken.

Each of the tokens forms a closed system, with its own circulation of coins, the consistency of which is ensured through the "key images" mechanism [2], this guarantees that the number of coins emitted by the token will remain constant. The commission in each transaction is paid by the native token, so any transaction that transfers synthesized tokens will also carry the native token.

2.3 Auditable wallets

Auditable Wallets is a technology that allows you to create wallets whose balances can be publicly tracked by providing certain information (tracking seed). Such wallets are used for proof of funds in various governance models, including Zano, for sharing information about the project fund. Tokenized transactions will also be able to take full advantage of Auditable Wallets, thus expanding the potential scope of Zano tokens.

2.4 Atomic Exchange via Consolidated Transactions

In Zano we've implemented HTLC-based atomic swaps, to introduce cross-chain exchange operations, but this mechanism has its own drawbacks. The HTLC contract scheme involves 3 phases, in which one of the parties always has a slight advantage over the other party (the party that makes the disclosure of the secret have the advantage of the final decision, and depending on the situation, can confirm or cancel the transaction, sometimes to the detriment of the other side). The Zano tokenization platform introduces Atomic Exchange operations that solve this problem. To do this, we use Zano Consolidated Transactions, a technology that allows two or more users to jointly and securely create one transaction that performs token exchange operations. This gives no advantage to any participants, since they all in the same position—theoretically anyone may withdraw operation by using inputs from exchange transaction in other transaction before exchange transaction got confirmed. Practically it would be quite complicated due to sporadic nature of block generation process.

As an example let's assume that Alice is willing to buy BTC with USD_X (let's call a token representing BTC as BTC_X). Alice creates a transaction and puts USD_X input from her wallet to transaction inputs. She should also include a BTC_X output addressed to her wallet to the transaction's outputs, so she makes sure that whoever use this half-created and half-signed transaction won't be able to use it without providing proper funding of the missing BTC_X part. It means that in this transaction template, which includes her part of assets, Alice has already defined conditions, on which she agrees to make this exchange operation, and it's not possible to make this transaction valid without fulfilling this conditions (Fig. 4).

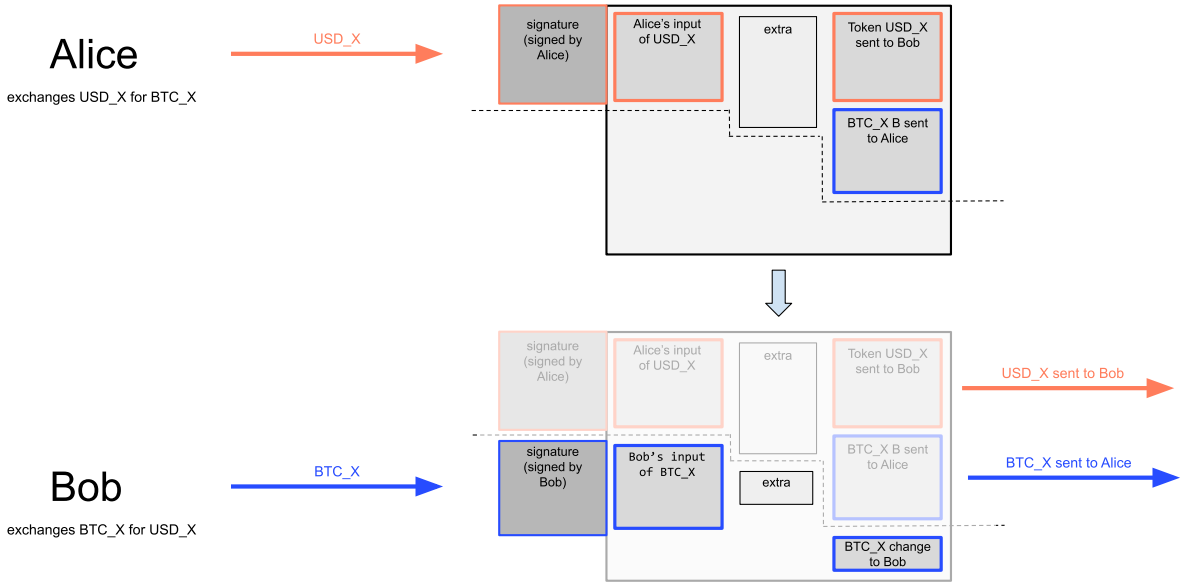

Fig. 4. Consolidated Exchange Transaction

We believe that this rapid and truly atomic exchange operations provide a solid basement for future tokenization ecosystem growth.

2.5 Private DEX on Zano

With Consolidated Transactions, we can provide safe and consistent way to perform exchange operations, first to make it work we need to enable on-chain orders matching. This part is covered by DEX Coordinator, an open-source server code, originally provided by Zano team. Coordinator's main feature is

ability to combine sell and buy orders together in an exchange transaction. Unlike CEX, Coordinator does not require registration or any other authentication. Private DEX user is paying small fee for order placement and maintenance (like gas fee in EVM-compatible networks). Provided fee used by Coordinator to host orders, handle updates, and deliver all necessary data to counterparty when order is matched. Coordinator doesn't take user's funds in custody; all assets are controlled by owners during entire process. Trustless setup of the Private DEX provides user with partially filled transaction for validation and co-signing to proceed with the trade. All transaction details remain private for outside observer, including total transaction volume, counterparty addresses and token details.

3 Token-aware RingCT transaction

3.1 Notation

Let $\mathbb{G}$ denote the main subgroup of the Ed25519 curve and let $\mathbb{Z}_p$ denote a ring of integers modulo p.

Let l be the order of $\mathbb{G}$: $l = \#\mathbb{G} = 2^{252} + 27742317777372353535851937790883648493$.

For any set X, $x \xleftarrow{\$} X$ means uniform sampling of x at random from X.

H_s is a scalar hash-function: $H_s : \{0,1\}^* \to \mathbb{Z}_l$

H_p is a deterministic hash function: $H_p : \{0,1\}^* \to \mathbb{G}$.

3.2 Basic idea

In a normal RingCT transaction each output's amount a is committed to in a commitment

$$A = aH + fG$$

where $H, G \in \mathbb{G}$ are group elements with no efficiently-computable discrete logarithm relation and $f \in \mathbb{Z}_l$ is a random blinding factor.

The initial emission of a token in Zano will be made through a transaction which pays (or burns) some amount of native coins and puts a public *token descriptor* with the following information associated with the given token:

- unique name, symbol, text description, etc.;
- emission parameters (like total or unlimited supply);
- owner pubkey;
- etc.

Let $t = H_s(token\_descriptor) \in \mathbb{Z}_l$ be the token's unique public scalar identifier. Consider a corresponding commitment for amount a and token t having the following form (hereinafter we omit output stealth address and encoded amount for simplicity):

$$A = aH + tX + fG$$

where $H, X, G \in \mathbb{G}$ are group elements with no efficiently-computable discrete logarithm relation and $f \in \mathbb{Z}_l$ is a random blinding factor. Then a typical transaction output would be represented by a tuple:

$$(\sigma^{rp}, A = aH + tX + fG)$$

Where σ^{rp} is a double-blinded range proof with respect to amount a (for instance, double-blinded extension of Bulletproofs+, suggested in [3]).

Consider a RingCT transaction with two inputs and four outputs ($m = 2, k = 4$). Each of input refers to a ring of three outputs ($n = 3$), and the second one is actually being spent.

#	Ring	Pseudo output commitment	Ring signature add. assertion	Outputs
	$(\sigma_0^{rp}, A_0 = ?)$		$A - A_0^p = ?$	$\underline{(\sigma_0'^{rp}, E_0 = e_0 H + t_4 X + y_0 G)}$
0	$(\sigma_1^{rp}, A_1 = a_1 H + t_1 X + f_1 G)$	$A_0^p = a_1 H + t_1 X + f_1' G$	$A - A_0^p = (f_1 - f_1')G$	$\underline{(\sigma_1'^{rp}, E_1 = e_1 H + t_4 X + y_1 G)}$
	$(\sigma_2^{rp}, A_2 = ?)$		$A - A_0^p = ?$	
	$(\sigma_3^{rp}, A_3 = ?)$		$A - A_1^p = ?$	$\underline{(\sigma_2'^{rp}, E_2 = e_2 H + t_1 X + y_2 G)}$
1	$(\sigma_4^{rp}, A_4 = a_4 H + t_4 X + f_4 G)$	$A_1^p = a_4 H + t_4 X + f_4' G$	$A - A_1^p = (f_4 - f_4')G$	
	$(\sigma_5^{rp}, A_5 = ?)$		$A - A_1^p = ?$	$(\sigma_3'^{rp}, E_3 = e_3 H + t_4 X + y_3 G)$

In this example input 0 corresponds to the native coin, and so output 2 (note using t_1). Input 1 corresponds to a token, associated with scalar t_4, and outputs 0, 1, and 3 use the same token-associated scalar t_4. Hence we need to prove that the following equations hold:

$$a_1 = e_2 + fee$$
$$a_4 = e_0 + e_1 + e_3$$

This can be done by proofing knowledge of DL secrets r_0, r_1 with respect to X for each different token:

$$A_0^p - E_2 - fee \cdot H = r_0 X$$
$$A_1^p - E_0 - E_1 - E_3 = r_1 X$$

or

$$\sum_{inputs(t)} A_i^p - \sum_{outputs(t)} E_j = r_i X$$

where $inputs(t)$ and $outputs(t)$ are inputs and outputs associated with the given token t. Note, that G-component is eliminated by proper choosing of blinding masks y_j.

3.3 Enforcing output commitments

To ensure the output commitments E_i have the correct form, the sender calculates a Schnorr-like proof of knowing DL secrets x_h, x_g of $A_i^p - E_j$ with respect to H, G:

$$A_i^p - E_j = x_h H + x_g G$$

for each output, where A_i^p is the source pseudo output commitment for the output E_j.

3.4 Size estimation

Let's try to estimate size of the all signatures and proofs for a transaction with m inputs (each having n elements in its ring) and k outputs.

Parameter	Elements in $\mathbb{G}$	Elements in $\mathbb{Z}_l$
Inputs (key images)	m	
Pseudo output commitments	m	
Ring signature (CLSAG)		$(n+1)m$
Outputs' range proofs (BP+)	$2 \cdot \lceil \log_2(64) + \log_2(k) \rceil + 3$	4
Outputs' additional proofs		$(2k+1)$ [1]
Outputs data (except proofs)	2	1
Total	$2m + 2\lceil \log_2(k) \rceil + 17$	$(n+1)m + 2k + 6$

Table 1: Signature size estimation

[1] 2-generators Schnorr proofs for each output with shared Fiat-Shamir challenge.

References

[1] Meni Rosenfeld. *Overview of Colored Coins*. https://bitcoil.co.il/BitcoinX.pdf. 2012.

[2] Nicolas van Saberhagen. *CryptoNote v 2.0*. https://cryptonote.org/whitepaper.pdf. 2013.

[3] sowle and koe. *Zarcanum: A Proof-of-Stake Scheme for Confidential Transactions with Hidden Amounts*. https://eprint.iacr.org/2021/1478.pdf. 2022.

[4] Peter Todd. *Stealth addresses*. http://www.mailarchive.com/bitcoin-development@lists.sourceforge.net/msg03613.html. 2014.