# ProjectEuanthe

# Stealth Address State Flotation Reduction.

Research Draft v1
draft@projecteuanthe.org
May 20, 2021

## 1 Introduction

ProjectEuanthe is a set of Decentralized Invisible Transaction (DIT) instances built on Ethereum. Euanthe deals with two modules namely ebb and elara, with distinguishable schemes leading to private transactions on the layer 1 Ethereum blockchain, through invisible addresses and shielded contract respectively. This paper deals with the solution to the problems of reducing search time in stealth address schemes focused on ebb's stealth address implementation.

## 2 Analysis

### 2.1 Security and Efficiency of ebb v1

In Ebb's DIT scheme, the system expects 32 byte random number for *r*, but can perform strongly with 16 bytes of data. The protocol uses elliptic curve cryptography and features its properties in this scheme.

The security provided by Elliptic Curve is equal to the half the size of the private keys of Ethereum, i.e., 16 bytes. Ebb stretches for maximum security. secp256k1 or Koblitz elliptic curve has some special properties that helps in performing group operation more efficiently.

In systems like ebb there exists a $O(N)/O(1)$ trade off, namely if a sender only modifies $O(1)$ number of bits, then recipient necessarily needs to perform a $O(N)$ scan to search for its match. It is possible to achieve $O(\log n)/O(\log 1)$ scan but it requires huge group elements. One of the solutions for overcoming such a trade-off is having separate spend and view keys and handing over the *pk_view and pub_Spend* keys to an oracle system that can compute the proof for $O(N)$ transactions and can inform recipients via push notifications for an incoming transaction.

### 2.2 Alternative Design Ideation for ebb

The following scheme takes into advantage the property of subgroup hiding assumptions in groups of unspecified order. This enables the system to hide a few bits of information that is sent to the public and the sender registers a true value to the recipient while registering false to all the other member of the anonymity set, {$n_d$ -> ***0*** }. This acts as a turnaround to avoid computing 64 bits of O(N) transactions, for each account.

**Setup.** Participant, *i* generates two safe primes, $d_i = 2d_i+1$ & $k_i = 2k_i+1$ and calculates $n_i = d_ik_i$ . The system wraps the result of, $|z_{ni}| = 4d_ik_i = 4n_i$ and randomly chooses $g_i$, $u_i$ of order $n_i$ in $Z_{ni}$. The user encodes $h_i = u_i^{ki} \bmod n_i$ and publishes his public key as ($n_i$, $g_i$, $h_i$).

**Counter Init.** Participant $i$ initializes a public register $S^i$ to $h_i^{ri}$ *mod* $n_i$ for a random value $r_i$ and sets the secret register $\omega_i$ to 1.

**Send.** A sender wants to transmit a secret bit $b$ to user $i$ :

- If $b$ = 1, he chooses random $a$ and computes $w_i = g_i^a \bmod n_i$ ;
- if $b$ = 0, he chooses random $b$ and computes $w_i = h_i^b \bmod n_i$ ;

The protocol then changes the user's public counter to $S_i = S_i . w_i \bmod n_i$ .

**Receive.** User $i$ computes $s_i = S_i^{di} \equiv g_i^{di\sum \mathrm{ldi}} \bmod n_i$ and checks if $s_i$ is equal to $\omega_i$ . If they're equal, the address then received only 0 bits from his last check. If not, he received at least one 1 bit and updates his secret register $\omega_i$ as $\omega_i = s_i$ .

**Counter Reset.** After receiving one or more 1 bit, each user $i$ could reset his public register by computing $w_i = S^{qi-1} \cdot h_i^{ri}$ for random $r_i$ and letting $S_i = S_i . w_i \bmod n_i$. He then resets his secret register $\omega_i$ to 1.

### 2.2.1 Spread Nullification

- The Chinese Remainder Theorem can be used to aggregate all the values $w_i$ to a single public value $w$, so that $w$ satisfies $w \equiv h_j^{bj} \bmod n_j$ for $j \neq i$ and $w \equiv g_i^a \bmod n_i$. In this case the counter updates' can be publicly performed by third parties, but the size of $w$ remains $O(M)$ where $M$ is the number of users in the anonymity set $z$.

- Senders possess the capability to privately modify the public registers, while other updates can be delegated by publishing an aggregated value $w$ on/off-chain. To hide senders' actions, some updates can be performed in the network like sending 0 bits to random users.

- To allow receivers know how many 1 bits value they received, the sender when sending the 1 bit to user $i$ could compute the value $w_i = g_i . h^b \bmod n_i$ , with $b$ random instead (the $h_i^b$ blinds the $g_i$ when the public counter is updated). In that case the receiver has to brute force $S_i^{di} \equiv g_i^{di\sum \mathrm{ldi}} \bmod n_i$ in base $g_i^{di}$ . It would be important to restrict the usage of no high ordered $g_i$ is used (i.e. all $a_l$ are at most 1).

- The Counter reset operation is optional and indistinguishable from a normal Send. If the receiver keeps tracks of how many 1 bit, he received a brute force could be performed on basis rounds.

- The secret register $\omega_i$ defines and underlying notion of time: for example, the receiver could regularly checks at intervals of k blocks his public register by checking if he received 1 or more 1 bits: if $S_i^{di} \bmod n_i$ is different from the last checked secret register $\omega_i$ , it means that some sender multiplied his public register with a random $n$ -order element and so there are new stealth transactions addressed to him in the last $k$ blocks.

# 3 Conclusion

This paper describes the architecture to successfully minimize end machine load bearings to carry out key pair matches through secret S from the ECDH. The Scheme for constructing this into the Ebb's DIT, maintains Receiver anonymity, due to the difficulty found in ECDLP. Cryptographic hash functions are used for updating the shared secret continuously for $n^{th}$ transaction. If adversary compromises a particular device and obtains $h_l$ of $l^{th}$ transaction, the adversary still won't be able to link previous transactions because of the inherent properties of the hash function.

# References


[EBB] draft@projecteuanthe.org "projecteuanthe/ebb: A modular DIT instance".

[FMD] Gabrielle Beck, Julia Len, Ian Miers, Matthew et al. "Fuzzy Message Detection".

[Ped] Gary Yu. "Blockchain Stealth Address Schemes".