Note Discovery Protocol - Request for Proposals

For key network decisions, we've been reaching outo to the community, seeking new ideas and proposals.

See also our Sequencer Selection RFP and Upgrade RFP.

We're now seeking ideas for efficient Note Discovery Protocols.

Overview

Private data within Aztec is stored within encrypted UTXOs or 'Notes'. These notes are then represented by their hash within the 'Note Hashes Tree'. In order for a user to consume a note that belongs to them, a mechanism needs to be in place to allow the note to be discovered. The lifetime of a note can be summarised as follows:

1. Alice executes a function that sends funds to Bob. A note is created containing the plaintext information describing the funds being given to Bob.
2. The note is then hashed and this hash is appended to the Note Hashes Tree.
3. Alice encrypts some or all of the note data to produce a ciphertext.
4. Encryption may be performed using a symmetric encryption key generated from a secret that Alice derives from a combination of Bob's public key and an ephemeral keypair that she generates solely for the purpose of encryption of this note.
5. Encryption may be performed using a symmetric encryption key generated from a secret that Alice derives from a combination of Bob's public key and an ephemeral keypair that she generates solely for the purpose of encryption of this note.
6. The ciphertext and the ephemeral public key generated by Alice are broadcast publicly to a Data Availability layer, as part of the process of building the rollup which incorporates Alice's function call.
7. Once the rollup has successfully settled on L1, Bob's funds can be spent. In order to do this he needs to: identify that he has a note; retrieve the note ciphertext; decrypt it to it's plaintext; and then identify the location of its hash in the Note Hashes Tree. Once Bob has successfully identified which note belongs to him, decryption is trivial. As is locating the index of the Note Hash in the Note Hashes Tree due to the deterministic way in which the note hashes are appended to the tree. That leaves the process of identifying which notes belong to Bob (Note Discovery) as the primary challenge that Bob faces when wishing to spend his funds.
8. Note Discovery can be performed naively. Bob could simply download every note ciphertext and associated ephemeral public key. For each note he can compute a secret using his secret key and generate the symmetric encryption/decryption key from that secret. If the note is indeed the one Alice created for bob then he will have derived the same secret and ultimately the required decryption key.

Whilst guaranteed to be successful and preserve privacy. This last step comes at signifcant cost to Bob in terms of the quantity of data he needs to download and the compute required to attempt decryption of every note. As the total number of all notes – and the rate at which notes are produced – increases over time, this method of trial-decryption becomes increasingly less feasible.

We are submitting this RFP in search of alternative methods of Note Discovery.

Problem statement

We consider 3 high level types of Aztec users with varying privacy tolerances, and varying willingness and/or ability to run trial decryption software. This RFP focusses on the "middle" group, who want a high degree of privacy, but do not want to run full nodes to perform brute force trial decryption of all UTXO notes.

User Type

Willingness / ability to run trial-decryption software

Required Privacy

Comment

Low resource, trusting.

No

Medium.

Happy with confiding secret info to a trusted server.

“Confidentiality” rather than privacy - private state is known by the trusted server, but not the general public.

Low resource, untrusting.

No

High

.

NOT happy with confiding secret info to a trusted server.

Happy interacting with an untrusted

server via privacy-preserving

requests.

This RFP focusses on this type of user.

We are happy to consider a range of solutions along the privacy/performance tradeoff space (see next section).

High resource

Yes

High.

Wants to discover notes on their own device, with no server interactions.

No new solution is needed - brute force decryption suffices.

Aztec is currently targeting up to 10 txs per second, generating on average 4 Notes of size 256 bytes.

That’s roughly 10 tx/s * (60 * 60 * 24 * 365) * 4 notes ~= 1.2 billion notes per year.

Note (haha): the size of a Note Plaintext (and hence of a Note Ciphertext) can vary in Aztec, because smart contracts are fully programmable. Note discovery solutions need to account for this variability, if designing privacy-preserving client-server protocols which discover and return ciphertexts back to a user.

Note also: a transaction can generate many more than just 4 notes; we have just provided “4” as a rough estimate of an average tx.

We do not expect users to always be online attempting to discover new notes on every new block. Consequently, proposed schemes will need to cater for users ‘catching up’ over a large number of blocks.

Requirements & Design Considerations

There are several factors for evaluating for the feasibility of a proposal:

1. Minimise: Client-side network upload/download size
2. Users can be assumed to have broadband with ~20mbps up/download.
3. Users can be assumed to have broadband with ~20mbps up/download.
4. Minimise: Client-side compute
5. Clients may need to process “hints” or decrypt false positives returned from the server.
6. Clients may need to process “hints” or decrypt false positives returned from the server.
7. Minimise: Sync time
8. Minimise: Additional on-chain data size (if any)
9. We may need to publish additional data to help users identify which notes are relevant to them. These will result in additional L2 (or L1) data publication costs, which must be paid in the note creator transaction.

10. We may need to publish additional data to help users identify which notes are relevant to them. These will result in additional L2 (or L1) data publication costs, which must be paid in the note creator transaction.
11. Minimise: The financial cost to the user
12. Maximise: Level of privacy
13. Some existing solutions leak a small amount of information for performance gains (e.g. FMD), but ideally leakages should be kept to a minimum.
14. Any solution should be resistant to transaction analytics.
15. Some existing solutions leak a small amount of information for performance gains (e.g. FMD), but ideally leakages should be kept to a minimum.
16. Any solution should be resistant to transaction analytics.
17. Minimise: Interactivity between sender and recipient.
18. To support note ownership “hints” (a.k.a. “clues” or “tags”), is there on-chain or off-chain communication that must happen between the sender and recipient?
19. If off-chain, does this introduce liveness issues or block discovery of notes from unknown senders? Is that acceptable?
20. If off-chain, does this introduce liveness issues or block discovery of notes from unknown senders? Is that acceptable?
21. Some solutions might require a ‘handshake’ to share a secret between sender and recipient.
22. To support note ownership “hints” (a.k.a. “clues” or “tags”), is there on-chain or off-chain communication that must happen between the sender and recipient?
23. If off-chain, does this introduce liveness issues or block discovery of notes from unknown senders? Is that acceptable?
24. If off-chain, does this introduce liveness issues or block discovery of notes from unknown senders? Is that acceptable?
25. Some solutions might require a ‘handshake’ to share a secret between sender and recipient.
26. Maximise: Snark efficiency
27. An app might wish to constrain the computation of a tag, and the encryption computation. So tagging & encryption should ideally be efficiently computable inside a snark.
28. An app might wish to constrain the computation of a tag, and the encryption computation. So tagging & encryption should ideally be efficiently computable inside a snark.
29. Protocol level changes
30. Solutions may involve modifications to the Aztec protocol itself (“enshrining”).
31. We may need to formalize a notion of epochs to chunk notes into smaller datasets for more-efficient private queries.
32. Solutions may involve modifications to the Aztec protocol itself (“enshrining”).
33. We may need to formalize a notion of epochs to chunk notes into smaller datasets for more-efficient private queries.

No solution will be dominant across all these, and we are open to solutions with different tradeoffs across these factors.

Submission Format

To ensure consistency and facilitate the review process, kindly adhere to the following submission format:

Title

: A concise, descriptive title for your proposal

Summary

: A brief summary of your proposal (150-300 words)

Details

: Explain the Note Discovery solution, any techniques/technologies it depends on and any tradeoffs made by it

Comparisons

: Explain what makes this solution unique and different from alternative solutions

Feasibility

: Explain the ability to implement this solution within the next 6-12 months

Questions

: Any outstanding questions

Submissions should be created as a new post on this forum, tagged note-discovery and rfp. Once the new post is created, please refer back to this RFP and post a short description + link your proposal.

Potentially helpful references

Cryptographic Techniques

Private Information Retrieval

Single Server PIR is a cryptographic protocol that allows a user to retrieve item(s) from a server such that the server learns nothing about which item was retrieved. A recent paper claims a new state of the art for batch query performance in this setting.

The server must preprocess the data and typically “hints” about the server’s content must be downloaded by the client; these tend to dominate server side compute costs in PIR.

Private Set Intersection

PSI is a multi-party computation technique to allow two (or more) parties to find the intersection of their sets without revealing any information about non-intersecting elements. A variant called Labeled PSI could be used to query key-value pairs stored on a server without the server learning which keys were requested (or what data was returned). In our case, it is acceptable for information on the server contents to leak, since the UTXO tree is already public.

This utilizes homomorphic encryption, which makes answering the queries more expensive for the server.

Oblivious Message Retrieval

OMR variants use homomorphic encryption and work on the principal of generating cryptographic clues along with the encrypted notes. A user needs to create and manage a “clue” key and “detection” key (both public).

For Alice to send a note to Bob, Bob will need to make his Clue Key available. Alice uses this key to create the clue and sends the clue with the transaction. Bob uploads his detection key to the OMR server and is then able to query for notes in private.

An benchmark for OMR was performed as part of Namada Taiga’s design discussion.

Fuzzy Message Detection

FMD would tag each note with a “fuzzy detection key”. Users then provide a detection key to a server, which allopws the server to identify which notes belong to the recipient, but also identifies non-matching notes with some pre-set false-positive rate. Depending on this false-positive rate, the server will only gain a ‘fuzzy’ understanding of which notes belong to the recipient.

A drawback of this approach is that it potentially leaks information through statistical analysis.

Probabilistic Data Structures

If notes are tagged with some metadata linked to their owner, we may be able to useprobabilistic data structures such as cuckoo filters to determine which epochs to query for notes, reducing client/server communication rounds. These were proposed for Bitcoin light client in BIP-0157 and BIP-0158.

Existing Approaches

ZCash

ZCash uses an optimized trial decryption, but are exploring smarter methods such asDAGSync. ZCash indexing is simpler

because of the uniform nature of their UTXOs, which only represent ZEC balances and are much more fungible than Aztec notes.

Trial decryption has been unfeasible since ZCash was DoS attacked by a large numbers of shielded transactions in 2022.

Taylor Hornby wrote an overview of the note discovery problem on the ZCash Security forum (along with personal opinions on the solution space).

Penumbra

Penumbra has chosen Sender FMD, a variant of Fuzzy Message Detection where the false positive rate is chosen by the sender rather than the receiver. Some discussion around this choice is available on github.

It is unclear whether there are statistical linkage attacks on their design.

Aztec Connect

Users were provided the entire history of UTXO notes and brute force decrypted them. This was possible due to the smaller transaction volume and short history of Aztec Connect, but resulted in 3-4 minutes laptop sync time towards the end of life, after ~1 year with ~3,800,000 notes (2 notes per transaction and 1,900,000 transactions).

Bitcoin Wallet Lookups

Blyss provides a service based on the Spiral PIR scheme for looking up bitcoin wallet balances privately. This is avoids note discovery, but provides a concrete use case of private information retrieval.

Submission Deadline

The deadline for submissions is Friday February 2nd 2024.

Grants

Complete proposals may be eligible for a retro-active cash grant and swag.

FAQ

We anticipate that you may have questions regarding the call for proposals. The following frequently asked questions and their corresponding answers should provide some clarification. Otherwise, feel free to post a question in the forum or follow us on Twitter for updates.

Q1. How will a proposal be chosen?

A1. Proposals will be evaluated based on their adherence to the requirements and design considerations, as well as the quality, feasibility, and innovation of the proposed solution. The selection committee, consisting of Aztec Labs employees and possibly external stakeholders, will determine the winning proposal and share the chosen solution publicly.

Q2. Who can submit proposals?

A2. Anyone!

Q3. Can I submit more than one proposal?

A3. Yes, you can submit multiple proposals if you have different ideas for note discovery.

Q4. What if my proposal does not fully meet the requirements?

A4. We still encourage you to submit your proposal and participate in the discussion, as your ideas could contribute valuable insights and help shape the final solution.

DISCLAIMER

The information set out herein is for discussion purposes only and does not represent any binding indication or commitment by Aztec Labs and its employees to take any action whatsoever, including relating to the structure and/or any potential operation of the Aztec protocol or the protocol roadmap. In particular: (i) nothing in these posts is intended to create any contractual or other form of legal relationship with Aztec Labs or third parties who engage with such posts (including, without limitation, by submitting a proposal or responding to posts), (ii) by engaging with any post, the relevant persons are consenting to Aztec Labs' use and publication of such engagement and related information on an open-source basis (and agree that Aztec Labs will not treat such engagement and related information as confidential), and (iii) Aztec Labs is not

under any duty to consider any or all engagements, and that consideration of such engagements and any decision to award grants or other rewards for any such engagement is entirely at Aztec Labs' sole discretion. Please do not rely on any information on this forum for any purpose - the development, release, and timing of any products, features or functionality remains subject to change and is currently entirely hypothetical. Nothing on this forum should be treated as an offer to sell any security or any other asset by Aztec Labs or its affiliates, and you should not rely on any forum posts or content for advice of any kind, including legal, investment, financial, tax or other professional advice.