Hello Anoma Community,

I'm Aki, and I am the co-founder ofMycel. We are currently developing a new approach called IARMM, which addresses the challenges of securely and efficiently processing user intents across different blockchain networks.

I would like to explore the potential for collaboration between Mycel's IARMM and the Anoma protocol. Anoma's chimera chain approach enables interoperability between chains with overlapping validator sets, while IARMM focuses on interoperability with non-overlapping networks like Bitcoin. I believe these approaches are complementary and could work together to expand the scope of cross-chain interoperability.

Acknowledgement: Thanks to @apriori connecting me with the Anoma community and providing me with the opportunity to engage in this forum.

Abstract

In the realm of intent-centric markets, where users express their desired trading actions and outcomes, securely and efficiently processing user intents across different blockchain networks poses significant challenges. The Interoperable Account-Resource Mapping Model (IARMM) introduces a novel approach to address these challenges by abstracting various blockchain models, such as the Account Model, UTXO Model, and Object Model, into a unified resource model. IARMM treats user intents as encrypted resources and leverages the concept of checks to enable secure and efficient intent processing across multiple blockchain networks.

The core concepts of IARMM include fragmented accounts for enhanced privacy, the resource machine for atomic and secure intent execution, and a unified resource representation for seamless cross-chain interactions. Through a use case diagram, we demonstrate how IARMM facilitates cross-chain asset transfers, showcasing its potential to revolutionize intent-centric trading by enabling secure, efficient, and interoperable execution of user intents in decentralized ecosystems.

The core concepts of IARMM include:

- Unified resource representation for seamless cross-chain interactions
- Censorship resistance, ensuring fair and equal treatment of all intents
- Batch settlement of intents, improving efficiency and reducing costs

[

image

2000×555 122 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/a1485128e08b654111b18b5244058bc35c752688.jpeg)

Motivation

Challenges in Processing Cross-Domain Intents

Intent-centric markets allow users to express desired actions and outcomes, such as trading assets, executing complex financial transactions, or participating in decentralized applications. However, processing intents securely and efficiently across different blockchain networks is challenging. It requires a framework that can handle cross-chain complexities, ensure execution integrity, and maintain user privacy.

Anoma's chimera chain approach enables interoperability between chains with overlapping validator sets, while IARMM focuses on interoperability with non-overlapping networks like Bitcoin. I believe these approaches are complementary and could work together to expand the scope of cross-chain interoperability.

IARMM addresses these challenges by abstracting diverse blockchain models into a unified resource model. This enables seamless cross-chain interactions and intent execution.

IARMM: The Check Analogy

IARMM draws inspiration from the concept of checks in the traditional financial system and adapts it to the decentralized trading landscape. In IARMM, user intents are represented as encrypted resources, similar to checks. Just as checks encapsulate the necessary information and instructions for a financial transaction, encrypted resources in IARMM encapsulate the details of the desired trading actions, such as the assets involved, quantities, and constraints.

When a user creates an intent, it is analogous to writing a check. The user specifies the desired trading parameters and encrypts them using homomorphic encryption techniques. The encrypted resource, or "check," is then mapped to a resource

machine, which manages the execution and settlement of the intent.

The resource machine acts as the bank in this analogy, responsible for processing the checks and ensuring the integrity of the transactions. It verifies the validity of the encrypted resources, performs the necessary computations and validations, and executes the intents accordingly.

Fragmented accounts, managed in ID, serve as the account numbers on the checks. They provide a way for users to securely identify and access their resources without exposing their main account information. Fragmented addresses add an extra layer of privacy and security to the intent execution process.

IARMM enables cross-chain trading by allowing intents to be processed and settled across multiple blockchain networks. The resource machine facilitates the secure and efficient execution of cross-chain transactions, eliminating the need for intermediaries and reducing trust requirements. It ensures that the intents are executed atomically, meaning that the entire transaction either succeeds or fails, preventing partial execution or loss of funds.

IARMM Architecture

The IARMM architecture consists of the following key components:

1. Account Fragmentation:
2. Users create fragmented accounts, represented as encrypted resources, to express their trading intents securely.
3. Each encrypted resource contains the desired assets, quantities, and constraints, can be protected using homomorphic encryption.
4. Fragmented accounts

enhance privacy and security by allowing users to manage their resources without exposing their main account information.

- Encrypted resources are mapped to the resource machine for secure processing and execution.
- Users create fragmented accounts, represented as encrypted resources, to express their trading intents securely.
- Each encrypted resource contains the desired assets, quantities, and constraints, can be protected using homomorphic encryption.
- Fragmented accounts

enhance privacy and security by allowing users to manage their resources without exposing their main account information.

1. Encrypted resources are mapped to the resource machine for secure processing and execution.
2. Mycel ID:
3. Mycel ID is a decentralized identifier for users' fragmented accounts, enabling secure and private management of resources.
4. It allows users to access and control their fragmented accounts without revealing their main account information.
5. Mycel ID serves as a secure authentication mechanism for users to interact with their fragmented accounts and execute intents.
6. Mycel ID is a decentralized identifier for users' fragmented accounts, enabling secure and private management of resources.
7. It allows users to access and control their fragmented accounts without revealing their main account information.
8. Mycel ID serves as a secure authentication mechanism for users to interact with their fragmented accounts and execute intents.

[

image

1920×1010 99.3 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/e1e35a8b198fb5ec7ced67475846566bdf0bab92.jpeg)

The IARMM architecture enables secure and efficient processing of user intents across multiple blockchain networks. The resource machine serves as the core component, abstracting the underlying blockchain models and facilitating the

execution of intents. Fragmented accounts, managed by Mycel ID, provide privacy and security to users, allowing them to manage their resources without exposing their main account information.

Account Fragmentation

Account fragmentation is a crucial component of the Account-Resource Mapping Model (IARMM) that enhances privacy and security in intent-centric trading. It allows users to manage their resources without exposing their main account information.

[

image

1920×1926 199 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/991432b034d76c539e4e3fb29e2b31eddb42c8ab.jpeg)

1. Threshold Server run by validators creates fragmented account:
2. The validators collaborate through a secure threshold cryptography to create a fragmented account
3. The fragmented account is created without revealing the private key to any individual validator (optional)
4. The validators collaborate through a secure threshold cryptography to create a fragmented account
5. The fragmented account is created without revealing the private key to any individual validator (optional)
6. User submits intents and deposits assets to fragmented account:
7. Users express their trading intents by submitting them to the application.
8. Users also deposit the necessary assets into their respective fragmented accounts.
9. Users express their trading intents by submitting them to the application.
10. Users also deposit the necessary assets into their respective fragmented accounts.
11. Application solves intents:
12. The application takes the submitted intents and solves them to determine the appropriate trades or actions to be executed.
13. This step may involve matching orders, resolving conflicts, and optimizing the overall execution.
14. The application takes the submitted intents and solves them to determine the appropriate trades or actions to be executed.
15. This step may involve matching orders, resolving conflicts, and optimizing the overall execution.
16. Create transaction:
17. The application creates a transaction based on the solved intents.
18. The transaction includes a list of pairs of hashes of the fragmented account public keys involved in the settlement.
19. The proofer (oracle, validator, or another entity) proves the balances of the fragmented accounts to ensure the validity of the settlement.
20. The application creates a transaction based on the solved intents.
21. The transaction includes a list of pairs of hashes of the fragmented account public keys involved in the settlement.
22. The proofer (oracle, validator, or another entity) proves the balances of the fragmented accounts to ensure the validity of the settlement.
23. Settle owner of the fragmented account:
24. The ownership of the fragmented accounts is settled based on the executed trades or actions.
25. The settlement process updates the ownership records in the fragmented accounts to reflect the new owners after the trades.
26. The ownership of the fragmented accounts is settled based on the executed trades or actions.

27. The settlement process updates the ownership records in the fragmented accounts to reflect the new owners after the trades.

Benefits of IARMM

IARMM offers several key benefits for intent-centric trading:

1. Atomic Cross-Chain Swaps:

IARMM enables atomic cross-chain swaps, ensuring that trades are executed in an all-or-nothing manner. If any part of the cross-chain swap fails, the entire transaction is rolled back, preventing partial execution and loss of funds.

1. Batch Settlement of Intents:

IARMM allows for the settlement of multiple intents in a single transaction, improving efficiency and reducing costs.

1. Enhanced Interoperability:

IARMM provides a framework for interoperability across diverse blockchain networks and asset types. Users can express their trading intents in a unified manner, regardless of the underlying blockchain architecture or asset standards.

1. Minimized Trust Requirements:

IARMM leverages cryptographic techniques like homomorphic encryption to minimize trust requirements.

1. Censorship Resistance:

IARMM ensures that validators cannot censor or discriminate against specific fragmented accounts (FAs). All valid intents are treated equally and executed fairly, regardless of the user or the specific FA involved.

IARMM revolutionizes intent-centric trading by providing a secure, efficient, and interoperable framework for processing user intents across multiple blockchain networks. Its minimized trust requirements, support for atomic cross-chain swaps, and enhanced interoperability make it a promising solution for the challenges faced in decentralized trading.

Key Generation

Mycel is a decentralized platform that aims to provide secure and private transactions while maintaining the integrity and verifiability of the system. One of the key components in achieving these goals is the use of threshold signatures, specifically the Flexible Round-Optimized Schnorr Threshold (FROST) signature scheme.

One of the primary reasons Mycel has chosen to employ FROST is its unique property of maintaining a fixed public key, even when the set of signers changes. In Mycel, users' assets are held in fragmented accounts (FAs), each associated with a unique public key. The security and ownership of these FAs are distributed among a group of validators using FROST.

Generate Key

Round1

Every validator V_i

samples t

random values a_{(i,j)}

and use these values coefficients define a degree t-1

polynomial f_i(x)

a_{(i,j)} \leftarrow \mathbb{Z_q} \forall 0 \leq t< n\

f_i(x)=a_{(i,0)} + a_{(i,1)}x+...+a_{(i, t-1)}x^{t-1}\ =\sum^{t-1}*{j=0} a*{(i,j)}x^j

Each V_i

computes a commitment polynomial F_i(x)

\phi_{(i,j)}=a_{(i,j)}G\ F_i(x)=\sum^{t-1}*{j=0}\phi*{(i,j)}x^i\ =f_i(x)G

Each V_i

computes a proof of knowledge to the corresponding a_{i,0}

by calculating \sigma_i

k_i

is a random nonce

\beta

is a context string unique to this DKG execution, which prevents replay attacks

k_i \leftarrow \mathbb{Z_q}\ R_i = k_iG \ c_i = H(i,\beta,\phi_{(i,0)}, R_i)\ s_i = k_i + a_{(i,0)} c_i \ \sigma_i = (R_i, s_i)

Each V_i

broadcast F_i(x) , \sigma_i

to all other validators though VoteExtension

**By encrypting this F_i(x)

with the user's public key using a homomorphic encryption, we can prevent validators from knowing the Fragmented Account's public key.

Upon receiving them, V_i

verifies F_j(x) , \sigma_l (1\leq j \leq n, j \neq i)

by checking

s_jG = R_j+c_j \phi_{(j,0)}

Round2

Each V_i

securely sends to each other validator V_j

a secret share (j,f_i(j))

which encrypted with validator public key.

And verifies their shares by calculating:

F_i(j)=f_i(j)G

Each V_i

calculates signing share s_i

by

s_i=f_1(i)+f_2(i)+...+f_n(i)\ =\sum^n_{j=1}f_j(i)

Each V_i

calculate their publish verification polynomial F(x)

and this will be stored with user id (Mycel ID)

F(x) =F_1(x)+F_2(x)+...+F_n(x)\ = \sum^n_{i=1}F_i(x)

Signing

Preprocess

Create \pi (1\leq j \leq \pi)

nonce pair (d_{ij}, e_{ij})\leftarrow \mathbb{Z}^_*q* *times* *mathbb{Z}^*_q

and commitment share (D_{ij}, E_{ij})

(D_{ij},E_{ij}) = (d_{ij}G, e_{ij}G)

Sign

Validator (SA) receives message m

from user and verify owner. then SA select t

validators (V_i

) and send them unused nonce pair B

and m

Each V_i

compute their response z_j

by

\rho_i=H(i,m,B)\ R=\prod D_j (E_j)^{\rho_j}\ c=H(R,Y,m)

z_i=d_i+(e_i\rho_i+\lambda_is_i c)

SA publish signature $\sigma=(R,z)$and each V_i

delete used nonce pair

Use Case: Cross-Chain Asset Trading

The IARMM module follows a modular design, allowing it to be integrated with various intent-centric applications and blockchain networks. This modular approach enhances interoperability and enables cross-chain trading regardless of where the intent-centric application is deployed.

The IARMM module is specifically responsible for settling the solved intents containing cross-chain orders. It abstracts away the complexities of cross-chain communication and ensures the secure and atomic settlement of trades across different blockchains.

[

image

2000×757 209 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/e54862a36a1c74c8df69d17751230321701a5bd9.jpeg)

Cross-Chain Trading Flow

1. Intent Expression

: Users express their trading intents by creating fragmented accounts on their respective blockchains and depositing the desired assets into these accounts. They specify the assets they wish to trade, the quantities, and any additional constraints.

1. Intent Solving

: The intent-centric DEX or solver application collects and matches the trading intents from users across different blockchains. It solves these intents and prepares the cross-chain orders for settlement.

1. Order Settlement

: The solved intents, containing the cross-chain orders, are passed to the IARMM module for settlement. The IARMM module interacts with the respective blockchains to ensure the atomic and secure settlement of the orders, guaranteeing that the trades are executed in an all-or-nothing manner.

1. Confirmation and Asset Transfer

: Once the orders are successfully settled, the IARMM module confirms the completion of the trades to the intent-centric application and the

users. The traded assets are securely transferred between the users' fragmented accounts on their respective blockchains.

Open Questions

- How can we facilitate the settlement of Bitcoin transactions through Mycel's IARMM via the Anoma P2P Layer?
- Where should the resource machine be executed? On the Anoma chain or the Mycel chain?

References

- Anoma Resouce Machine https://zenodo.org/records/10498991
- Typhon's Chimera Chains Typhon's Chimera Chains | Blog - Anoma
- FROST: Flexible Round-Optimized Schnorr Threshold Signatures FROST: Flexible Round-Optimized Schnorr Threshold Signatures
- Modifying FROST Signers and Threshold Modifying FROST Threshold and Signers · GitHub

I am excited to engage in open discussions, explore the possibilities of intent-centric markets, and collaborate on advancing cross-chain interactions through this forum. Your insights and feedback are highly valued and appreciated.

Thank you for your attention, and I look forward to the discussions ahead.