Zero-Knowledge Proofs and OAuth 2.0 for Anonymity and Security in Distributed Systems

Ayman Nait Cherif[1*], *Youssef* Achir[1], *Mohamed* YOUSSFI[1], *Mohamed* YOUSSFI[1], *Mouhcine* ELGAREJ[1], *Omar Bouattane*[1]

[1]2IACS Laboratory & Adria B-T, ENSET, Hassan II University, Casablanca, Morocco
[2]EESS Laboratory, ENSET, Hassan II University, Casablanca, Morocco

Abstract. Abstract—This paper investigates the integration of Zero-Knowledge Proofs (ZKP) and OAuth 2.0 to enhance anonymity and security in multi-agent distributed systems. We propose an approach that allows agents to authenticate and prove possession of specific data without revealing the underlying details. Additionally, we outline a potential access control strategy using ZKP for anonymity, allowing agents to validate their access rights without identity exposure. When combined with OAuth 2.0, this mechanism provides a framework for secure data access. While the proposed methods offer promising solutions to security, privacy, and anonymity challenges in multi-agent systems, they also highlight the need for further research and validation to confirm their effectiveness.

1 Introduction

Multi-agent systems (MAS) have been a topic of interest since the late 20th century, with the first formal theories emerging in the 1970s [1]. These systems, composed of multiple interacting agents, offer a powerful and flexible approach to solving complex problems that are beyond the capabilities of individual agents. These methods have found successful applications across diverse areas such as robotics, artificial intelligence, economics, and computer science.

However, the distributed nature of MAS presents unique challenges, particularly in terms of security and anonymity. In a multi-agent system, agents often need to share sensitive information to achieve their goals. This information exchange can be vulnerable to various security threats, including eavesdropping, data tampering, and unauthorized access. For instance, an adversary could potentially intercept the communication between agents and gain access to sensitive information. This not only compromises the confidentiality of the data but also the integrity of the system.

Moreover, the anonymity of agents is another significant concern in multi-agent systems. In certain scenarios, it is crucial for agents to keep their identities hidden while interacting with others. For example, in a bidding system, agents might want to keep their bids anonymous to prevent other agents from manipulating the bidding process. However, traditional

* Corresponding author: ayman.nait.cherif@gmail.com

authentication methods often require agents to reveal their identities, thus compromising their anonymity [2].

In recent years, Zero-Knowledge Proofs (ZKP) and OAuth 2.0 have emerged as promising solutions to these challenges. ZKP, a cryptographic method, allows an agent to prove that they possess certain information without revealing the information itself, thus maintaining their anonymity. OAuth 2.0, on the other hand, is an authorization framework that enables third-party applications to obtain limited access to an HTTP service, which can be used to secure data access [4].

Despite the advantages of OAuth 2.0 in securing data access, it does not inherently provide anonymity. The issuing server can still associate access tokens with the specific agents to whom they were issued. This could potentially compromise the anonymity of the agents in the system. However, the integration of Zero-Knowledge Proofs (ZKP) can address this limitation. Zero-Knowledge Proof (ZKP) is a cryptographic technique that enables an entity to demonstrate ownership of specific information without disclosing the information, thereby preserving anonymity [5].

Fundamentally, the applicability of OAuth 2.0 in a multi-agent system hinge on the needs of that system. Should the system demand token-based authentication, delegated authorization, and a protocol that is both standardized and broadly supported, OAuth 2.0 might be an appropriate choice. However, if the system also requires anonymity, additional measures, such as the integration of ZKP, would need to be taken.

This paper proposes a model that combines ZKP and OAuth 2.0 to enhance security and anonymity in MAS. We believe that this approach can address the limitations of existing methods and provide a robust framework for secure, anonymous communication in distributed systems. The primary goal of this paper is to investigate the capabilities of this model and to offer an in-depth examination of how it is implemented and its efficiency.

2 Related Work

The integration of Zero-Knowledge Proofs (ZKP) and OAuth 2.0 has been explored in various studies. A study by Fotiou et al. [6] introduced a new kind of OAuth 2.0 token supported by a distributed ledger. This design, facilitating proof-of-possession, auditing, and accountability, utilizes smart contracts to offer enhanced token management services like revocation, delegation, and equitable exchange. A proof-of-concept for their solution was implemented using Ethereum smart contracts and adhering to the ERC-721 token specification. This approach is innovative in its use of blockchain technology to enhance the security and functionality of OAuth 2.0 tokens. However, their approach does not consider multi-agent systems and does not provide anonymity for agents. This leaves room for further research and development in this area, particularly in the context of multi-agent systems where anonymity can be a crucial requirement.

In a different research, Li et al. [7] presented a secure capability-based system accommodating "permission sequence" and "context." This structure permits a limited series of permissions to be applied, each within its unique context. The authors validated the safety attribute of this system under these circumstances and incorporated it into OAuth 2.0 using proof-of-possession tokens. This work is significant in its introduction of a context-aware permission sequence, which can provide a more granular and context-sensitive access control mechanism. However, although this research offers a strong structure for implementing permissions, it overlooks the matter of anonymity in multi-agent systems. This indicates a potential area for further research, where the concepts of context-aware permission sequences and anonymity could be combined to provide a more secure and privacy-preserving access control mechanism in multi-agent systems.

TrustMAS [8] is a platform designed to guarantee trust and anonymity for mobile agents. It employs an anonymous method utilizing a random-walk algorithm to facilitate general-purpose anonymous communication among agents. This approach is unique in its focus on mobile agents and its use of a random-walk algorithm for anonymity. However, TrustMAS does not integrate Zero-Knowledge Proofs or OAuth 2.0, limiting its application in certain scenarios. This suggests a potential area for further research, where the concepts of Zero-Knowledge Proofs and OAuth 2.0 could be integrated into the TrustMAS platform to enhance its security and functionality.

Alderman, J., & Crampton, J. [9] investigated the application of Key Assignment Schemes (KASs) in entity authentication protocols. They regarded the communicating entities as autonomous agents engaging repeatedly through their private decision models. Decisions regarding authentication and key agreement are made according to the agents' behaviors observed during the interaction. This approach is interesting in its use of autonomous agents and their observed behaviour for authentication and key agreement. However, this approach does not explicitly incorporate Zero-Knowledge Proofs or OAuth 2.0. This indicates a potential area for further research, where the concepts of Zero-Knowledge Proofs and OAuth 2.0 could be integrated into this approach to enhance its security and functionality.

A novel approach to authentication and key agreement that is secure against quantum computing was proposed by [10]. This method considers the communicating entities as independent agents that engage with one another repeatedly, utilizing their individual decision-making models. While this approach provides a robust framework for authentication and key agreement, it does not address the issue of anonymity in multi-agent systems. This suggests a potential area for further research, where the concepts of anonymity could be integrated into this approach to provide a more secure and privacy-preserving authentication and key agreement mechanism in multi-agent systems.

ANAP [11], a new anonymous authentication protocol for mobile ad hoc networks (MANETs) augmented with a distributed reputation system, introduces mechanisms that mask the genuine identity of the communicating nodes and can counteract familiar attacks. However, ANAP does not integrate Zero-Knowledge Proofs or OAuth 2.0, limiting its application in certain scenarios. This indicates a potential area for further research, where the concepts of Zero-Knowledge Proofs and OAuth 2.0 could be integrated into the ANAP protocol to enhance its security and functionality.

3 Background

Multi-Agent Systems (MAS) consist of several interacting agents within a computerized framework. They aim to address challenges that are too complex or unattainable for a singular agent or a unified system to tackle. In these systems, the agents are autonomous and engage with one another, either collaborating to attain a shared objective or competing to achieve individual goals. The decentralized character of these systems introduces distinct challenges, especially concerning security and anonymity.

In more detail, MAS is a significant area of research in distributed computing. They offer a way to model complex systems composed of autonomous entities, allowing for more efficient problem-solving and decision-making. The agents in a MAS can be physical entities, such as robots, or virtual entities, such as software programs.

However, the distributed nature of MAS presents unique challenges, and among them, ensuring secure and anonymous communication between agents stands out as a significant concern. In a MAS, agents often need to exchange sensitive information to achieve their goals, be it collaboration, negotiation, or competition. The information exchanged may include strategic data, private preferences, or confidential credentials. Ensuring the security of this information and the anonymity of the agents involved becomes paramount, not merely

as a matter of privacy but as a fundamental requirement for the correct and trustworthy operation of the system. A breach in security or a failure in maintaining anonymity can lead to a wide array of problems, ranging from unauthorized access to critical system functions to the exposure of sensitive agent behavior. This challenge extends beyond traditional security measures and requires innovative cryptographic methods.

Zero-Knowledge Proofs (ZKP), a groundbreaking cryptographic technique, comes into play here, providing a robust solution to this intricate problem by allowing agents to prove the validity of a statement without revealing any information about the statement itself. The application of ZKP in MAS is a promising avenue to address this complex challenge, opening new possibilities for secure and anonymous multi-agent interactions.

Zero-Knowledge Proofs (ZKP) is a cryptographic method that allows an agent to prove that they possess certain information without revealing the information itself. This idea is vital for preserving the anonymity of agents within a system. In the framework of our suggested model, ZKP is employed to enable agents to authenticate themselves and demonstrate ownership of information without disclosing the details of that data. By utilizing ZKP, agents can generate proof that their message is consistent with a known hash, ensuring the integrity of the message while preserving anonymity.

ZKP is an intriguing principle in cryptography with substantial effects on privacy and security. It was initially brought forth in the 1980s by Shafi Goldwasser, Silvio Micali, and Charles Rackoff [5]. The underlying concept of ZKP is that it permits one entity (the prover) to demonstrate to another (the verifier) that they are aware of a value x, without divulging any information other than the fact that they know the value x. This is accomplished through a challenge-and-response mechanism, where the verifier issues a challenge to the prover, and the prover replies in a manner that confirms they know x without exposing it.

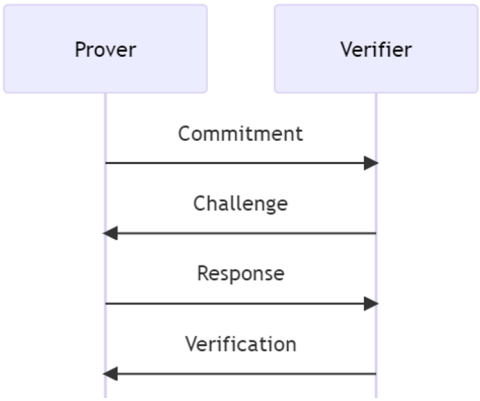

Fig 1: Zero Knowledge Proof workflow

Conversely, OAuth 2.0 is an authorization framework that allows third-party applications to gain restricted access to an HTTP service. In our suggested model, it's employed to safeguard data access. OAuth 2.0 facilitates a procedure for end-users to grant third-party access to their server resources without divulging their credentials (usually a username and password pair), utilizing user-agent redirections. However, OAuth 2.0 does not inherently provide anonymity as the issuing server can still associate access tokens with the specific agents to whom they were issued. This could potentially compromise the anonymity of the agents in the system.

OAuth 2.0 is a commonly used protocol that enables applications to protect user data while offering a smooth user experience. It creates a pathway for clients to access server resources on behalf of a resource owner, like an end-user. This protocol is frequently employed to grant third-party applications access to HTTP services, either on behalf of the resource owner or by permitting the third-party application to gain access independently.

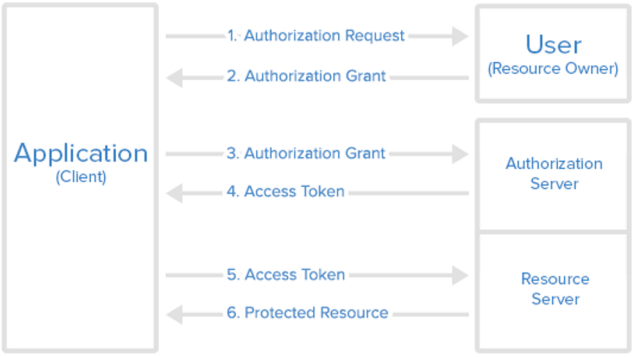

Fig 2: OAuth 2 abstract protocol flow

In our proposed model, we integrate ZKP with OAuth 2.0 to address this limitation. The combination of these two technologies allows for a secure framework for data access while maintaining the anonymity of the agents. This mechanism offers promising solutions to the challenges of security, privacy, and anonymity in multi-agent distributed systems. However, further research and testing are needed to validate their effectiveness.
The integration of ZKP and OAuth 2.0 is a novel approach in the realm of distributed systems. By combining the robust security and privacy features of ZKP with the widespread adoption and ease of use of OAuth 2.0, we aim to create a system that not only ensures secure data access but also maintains the anonymity of the agents involved. This could have far-reaching implications for a wide range of applications, from secure multi-agent systems to privacy-preserving data sharing and beyond.

4 The proposed model

In distributed computing, the decentralized structure of Multi-Agent Systems (MAS) brings forth certain security considerations. This chapter offers an exploration of a model that seeks to enhance security and anonymity within MAS. By weaving together cryptographic techniques with specific servers and modules, the intention is to present a potential approach to some of the challenges MAS face. As we move forward, we'll delve into the various components and workflows, hoping to provide a clearer understanding of agent communication in such systems.

4.1 System components and architecture

The proposed model for secure and anonymous communication within Multi-Agent Systems (MAS) is comprised of several integral components. Each component plays a specific role in ensuring the functionality, security, and efficiency of the system.

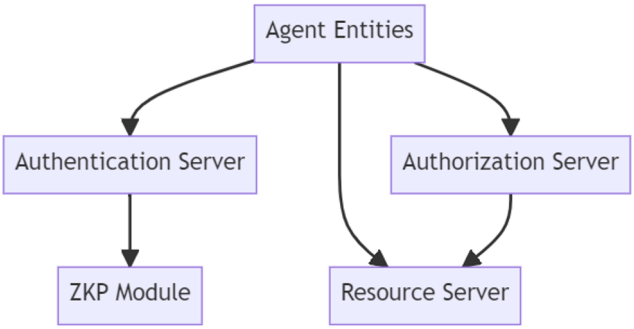

Fig 2: System components

Below, we elaborate on these components:

- **Agent Entities**: are the primary actors within the MAS. Every agent possesses a unique identifier and can assume various roles, including that of a requester, provider, or verifier. Agents are responsible for initiating and responding to requests, performing tasks, and interacting with other components within the system. They are the building blocks of the MAS and their behaviour is governed by predefined rules and protocols.
- **Authentication Server**: is tasked with confirming the identity of agents. It uses cryptographic techniques, including Zero-Knowledge Proofs (ZKP), to authenticate agents without revealing their actual credentials. This server acts as a gatekeeper, ensuring that only authorized agents can access the system's resources.
- **Authorization Server**: manages access permissions within the system. It employs OAuth 2.0 protocols to grant tokens that agents can use to access specific resources. The server defines the scope of access for each agent and ensures that they only access resources for which they have the necessary permissions.
- **Resource Server**: stores and manages the resources that agents might want to access. These resources could include data, services, or computational resources. The server ensures that resources are available to authorized agents and maintains the integrity and confidentiality of the data.
- **ZKP Module**: is a dedicated component that facilitates the creation and verification of zero-knowledge proofs. It interacts closely with the Authentication Server to provide an additional layer of security. The module ensures that agents can prove possession of specific information without revealing the information itself.

The system components of the proposed model are meticulously designed to cater to the unique requirements of MAS. The combination of cutting-edge cryptographic techniques with specialized servers and modules ensures that agents can operate in a secure, efficient, and anonymous environment. The modular nature of these components allows for flexibility and scalability, making the system adaptable to various use cases and scenarios. By understanding the roles and functionalities of these components, we can further explore the workflows and integrations that make up the complete system in the subsequent sections.

4.2 Agent Identification and Authentication Workflow

4.2.1 Integration of Zero-Knowledge Proofs (ZKP) Authentication

Zero-Knowledge Proofs (ZKP) offer a way for one party to demonstrate to another that a specific statement is true, without revealing any extra information. Within the framework of Multi-Agent Systems (MAS), ZKP can be utilized to authenticate agents securely and anonymously.

a. Setup phase

The setup phase is a critical part of the ZKP authentication process. During this phase, the following public parameters are predefined:

- **Parameter p:** A large prime number that defines the finite field in which the computations are performed. It serves as the modulus for mathematical operations.
- **Parameter g**: A generator of the multiplicative group of integers modulo p. It's used to generate public and private keys.
- **Parameter h**: A value derived from the generator g and the secret value. It's used in the proof construction. It can be computed as $h = g^x \ mod \ p$, where x is a secret value.

The predefined parameters are then used by all the agents in the system that are involved in the ZKP authentication process. The parameters are considered public and are known to all parties in the system.

In the context of the proposed architecture, these parameters would likely be predefined during the initial setup phase of the system, either by the protocol designer or the system administrator. They would be stored in a secure and accessible location so that all agents involved in the ZKP authentication process can access them when needed.

b. ZKP Authentication Workflow

1. **Commitment to the Secret**: The prover commits to the secret they wish to prove they know without revealing it. For example, the prover might compute the commitment as: $C = g^s . h^r \ mod \ p$ where s is the secret, r is a random number, and C is the commitment.
2. **Challenge**: The verifier issues a challenge to the prover. The challenge ensures that the prover must respond in a way that is dependent on the secret.
3. **Response**: The prover responds to the challenge by calculating $z = \mathrm{r} + \mathrm{c.s} \ mod \ (p - 1)$. This response proves they know the secret without revealing it.
4. **Verification**: The verifier evaluates the prover's response by calculating $g^z . h^{-c} \ mod \ p = \mathrm{C}$.

If the equation holds, the verifier believes that the prover is aware of the secret, without uncovering its actual value.

This process ensures that the prover can authenticate themselves without revealing the specific details of their secret information. It provides a robust and secure method for agent authentication in distributed systems.

By integrating ZKP into the MAS, the system can ensure secure and anonymous authentication of agents, addressing one of the unique challenges presented by the distributed nature of MAS.

4.3 OAuth 2.0 Integration and ZKP

4.3.1 Introduction

Zero-Knowledge Proofs (ZKP) are renowned for their ability to enable secure and private authentication. However, when used alone in a Multi-Agent System (MAS), they present certain limitations:

- **Performance Issues**: ZKP can be computationally intensive, especially in large-scale systems. In a system with numerous agents, the computational overhead of ZKP can lead to delays.
- **Limited Functionality**: While ZKP provides robust authentication, it lacks comprehensive authorization mechanisms. For example, it doesn't handle token management or access control, essential components for secure communication between agents.

Integrating OAuth 2.0 with ZKP offers a synergistic solution that leverages the strengths of both technologies:

- **Enhanced Performance**: By utilizing OAuth 2.0's efficient token management system, the combined approach can mitigate the computational overhead of ZKP, improving overall system responsiveness.
- **Comprehensive Security**: OAuth 2.0 provides a standardized framework for authorization, complementing ZKP's privacy-preserving authentication. This integration ensures a complete and robust security solution.
- **Scalability and Flexibility**: The combination of OAuth 2.0 and ZKP offers a scalable and flexible solution that can be adapted to various system configurations and requirements.
- **Reduced Complexity**: While ZKP alone can be intricate, integrating it with OAuth 2.0 provides a structured approach that simplifies implementation. The standardized OAuth 2.0 framework guides the integration, reducing potential errors and complexities.

By combining the privacy assurance of ZKP with the well-established OAuth 2.0 framework, the proposed approach offers a superior solution for agent identification and authentication in MAS, addressing the challenges of using ZKP alone.

4.3.2 Workflow

1. **Agent Requests Authentication:** The agent begins the authentication process by selecting a random value r . This value is used to create a commitment to a secret. The agent computes the commitment $C = g^r \ mod \ p$ using the random value and public parameters g and p. This action binds the Agent to the secret without revealing it. The agent then sends the commitment C to the Authorization Server, establishing the Agent's intent to authenticate.
2. **OAuth 2.0 Authorization Request (Agent & Authorization Server):** The Agent initiates the OAuth 2.0 process and integrates the Zero-Knowledge Proof (ZKP) by including the commitment. The Authorization Server validates the request and associates the commitment with the specific OAuth 2.0 authorization process.
3. **Challenge from Authorization Server (Authorization Server & Agent):** The Authorization Server generates a random challenge C to test the Agent's knowledge of the secret. The agent receives the challenge and prepares to prove its knowledge of the secret without revealing it.

4. **ZKP Response (Agent & Authorization Server):** The Agent calculates the response $z = \mathrm{r} + \mathrm{c.s}$, using the random value, challenge, and secret. The response is then sent to the server for verification.
5. **Verification and Token Issuance (Authorization Server):** The server verifies the Agent's response using the commitment and challenge $g^z \equiv C.h^c \ mod\ p$, where $h = \ g^s \ mod\ p$. If the verification is successful, the server issues an authorization code, a temporary credential that the Agent can exchange for an access token.
6. **Agent Exchanges Authorization Code for Access Token:** The Agent sends the Authorization Code to the Authorization Server. The server validates the code and issues an access token, enabling the Agent to access protected resources.
7. **Accessing Resources (Agent & Resource Server):** The Agent uses the token to request access to resources on the Resource Server. The Resource Server validates the token and, if valid, grants access to the requested resources.

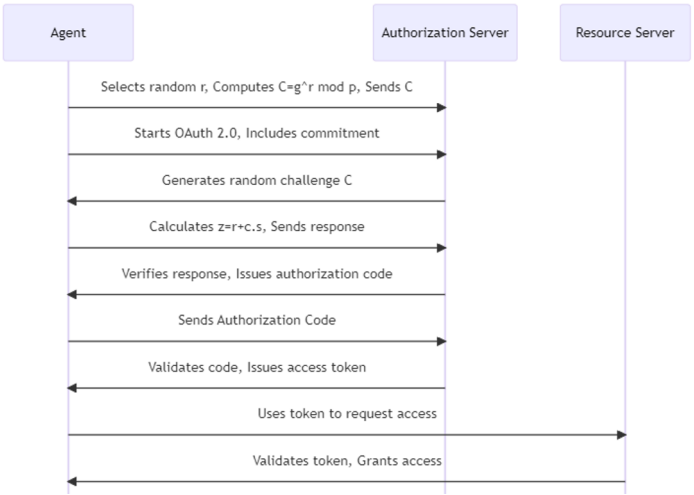

Fig 3: ZKP and OAuth 2 workflow

This workflow illustrates the seamless integration of Zero-Knowledge Proofs (ZKP) with OAuth 2.0, providing a robust and efficient solution for agent authentication in multi-agent systems (MAS). By combining the strengths of both technologies, the proposed approach offers enhanced security, and flexibility, addressing the limitations of using ZKP alone.

In this workflow, the ZKP commitment, challenge, and response are used to authenticate the Agent, and the OAuth 2.0 process is used to issue an access token, ensuring that the Agent is authenticated using ZKP before receiving the OAuth 2.0 token. The authorization code serves as a bridge between the ZKP authentication and the OAuth token issuance, confirming this authentication. The subsequent exchange for the access token ensures that the authenticated Agent is the one accessing the resources.

Through the utilization of ZKP, the Agent can demonstrate awareness of a secret without disclosing it, and the OAuth 2.0 procedure guarantees a standardized and secure method for granting access to resources.

4.3.3 Agent Communication

Agent communication is a vital aspect of the proposed architecture, leveraging Zero-Knowledge Proofs (ZKP) and OAuth2. The architecture's communication needs can be met through various methods, each with its unique characteristics and potential challenges. While traditional approaches like Direct HTTP Requests, WebSockets, and Peer-to-Peer

Connections have been widely used, they present certain limitations in terms of scalability, complexity, and control. Here's a closer look at these methods:

1. Traditional Communication Methods:
    - **Direct HTTP Requests**: Synchronous and blocking, leading to potential delays and scalability issues.
    - **WebSockets**: Allows real-time communication but can be complex to manage and lacks built-in message persistence.
    - **Peer-to-Peer Connections**: Can be efficient but requires complex network management and lacks centralized control.
2. Integration of Message Queues:

Transitioning from traditional communication methods, the integration of Message Queues emerges as a more flexible and robust solution. This approach offers several key advantages:

- **Asynchronous Communication**: Enables non-blocking communication, allowing agents to continue with other tasks without waiting for a response.
- **Scalability**: Supports horizontal scaling to handle many concurrent communications.
- **Message Persistence**: Ensures that messages are not lost and can be consumed later if the receiving agent is not available.
- **Integration with ZKP and OAuth2**: Seamlessly integrates with the authentication and authorization workflow.

3. Example Workflow with Message Queues

The integration of ZKP and OAuth2 with Message Queues adds another layer of security and functionality to the architecture. This integration is detailed as follows, enhancing the overall communication process:

- **ZKP Authentication**: Agents authenticate using ZKP, ensuring secure and anonymous identity verification.
- **OAuth2 Authorization**: Post-authentication, agents obtain an OAuth2 token, granting them access to specific queues or topics.
- **Message Publishing and Consumption**: Agents can publish and consume messages asynchronously, using the OAuth2 token for authorization. End-to-End Security: Messages can be encrypted and signed, ensuring confidentiality and integrity.

An example of implementing this approach might include the following steps. It illustrates the practical application of these concepts, providing a clear pathway for execution:

- **Agent Authentication and Authorization**: Agents authenticate using ZKP and obtain an OAuth2 token.
- **Message Publishing**: An agent publishes a message to a specific queue, using the OAuth2 token for authorization.
- **Message Consumption**: Another agent subscribes to the queue and consumes the message, processing it as required.
- **Asynchronous Workflow**: All communication is non-blocking, allowing agents to perform other tasks concurrently.

In conclusion, while other communication methods have their use cases, the integration of Message Queues in the architecture using ZKP and OAuth2 offers a robust and scalable solution for non-blocking asynchronous communication.

This approach ensures efficient handling of concurrent communications, integrates seamlessly with the secure authentication and authorization workflow, and provides a comprehensive framework suitable for modern distributed systems.

The asynchronous nature of Message Queues, combined with the security of ZKP and the flexibility of OAuth2, creates a powerful communication paradigm that aligns well with the needs of complex, distributed architectures.

5 Conclusion

The study conducted on secure and anonymous communication within Multi-Agent Systems (MAS) has revealed both promising avenues and real-world challenges. Through the exploration of existing technologies like OAuth 2.0 protocols, the research aimed to create a practical yet safe environment for agents to communicate.

The work presented here may be considered a step toward a more comprehensive solution. It introduces some innovative ideas but acknowledges that there are limitations in complexity and scalability. While the combination of cryptographic techniques seems promising, it may require further refinement to be applicable across various domains.

What's valuable in this study is its attempt to marry traditional methods with newer technologies to address long-standing problems in MAS. However, there are still questions that remain unanswered, such as the full optimization of the proposed system.

Future directions might include the exploration of different communication paradigms or further examination of cryptographic techniques. This research has opened the door to these possibilities, but it is clear that the path ahead is still long and filled with opportunities for exploration and learning.

In conclusion, this study offers a modest addition to the domain of secure communication within MAS. It invites further inquiry and collaboration, offering a starting point rather than a final answer. It's a reminder that even in the ever-advancing field of technology, there's always room for growth, improvement, and, most importantly, continued curiosity.

6 References

1. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," in IEEE Transactions on Computers, vol. C-29, no. 12, pp. 1104-1113, Dec. 1980, doi: 10.1109/TC.1980.1675516.
2. Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2), 84-88.
3. Zhu, Xiaoyang & Badr, Youakim. (2018). Identity Management Systems for the Internet of Things: A Survey Towards Blockchain Solutions. Sensors. 18. 4215. 10.3390/s18124215.
4. Hardt, D. (2012). The OAuth 2.0 authorization framework. Internet Engineering Task Force (IETF).
5. Goldwasser, Shafi, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof-systems." SIAM Journal on Computing 18.1 (1989): 186-208.
6. Fotiou, N., Pittaras, I., Siris, V. A., Voulgaris, S., & Polyzos, G. C. (2021). OAuth 2.0 Authorization using Blockchain-based Tokens.
7. Li, A. S., Safavi-Naini, R., & Fong, P. W. L. (2022). A Capability-based Distributed Authorization System to Enforce Context-aware Permission Sequences. Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT.
8. Szczypiorski, K., Margasiński, I., Mazurczyk, W., Cabaj, K., & Radziszewski, P. (2008). TrustMAS: Trusted communication platform for multi-agent systems. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5332 LNCS(PART 2)

9. Alderman, J., & Crampton, J. (2013). On the use of key assignment schemes in authentication protocols. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7873 LNCS.
10. Ahmed, I. H., Hanna, J. P., Fosong, E., & Albrecht, S. v. (2021). Towards Quantum-Secure Authentication and Key Agreement via Abstract Multi-Agent Interaction. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12946 LNAI.
11. Ciszkowski, T., & Kotulski, Z. (2006). ANAP: Anonymous Authentication Protocol in Mobile Ad hoc Networks. In X Krajowa Konferencja Zastosowań Kryptografii ENIGMA 2006.