electronics

MDPI

Article

A Three-Layer Authentication Protocol for Secure Access Control in AI-Based Enterprise Document Management Systems

Hyun-A Park

Department of Computer Engineering, College of AI Convergence, Honam University, 112, Honamdae-gil, Gwangsan-gu, Gwangju 62399, Republic of Korea; kokokzi@honam.ac.kr; Tel.: +82-62-940-3655

Abstract

AI-based enterprise document management systems increasingly integrate semantic retrieval, Retrieval-Augmented Generation (RAG), multimodal interaction, and data analytics. In such environments, protected objects extend beyond original documents to chunks, embeddings, retrieval outputs, generated responses, and analytical results. Consequently, conventional login-based or static role-based access control is insufficient. This paper proposes a three-layer authentication protocol that sequentially performs session authentication, role/permission verification, and event-based group-membership authentication. Unlike conventional group-key approaches, the proposed method uses event-specific group keys only as authentication material, not for document re-encryption. By combining forward and reverse hash chains, the protocol supports group key secrecy, forward secrecy, and backward secrecy in dynamic group environments. Security analysis shows that the protected authentication values are computationally indistinguishable from random bit strings of the same length under the stated assumptions. In addition, the efficiency analysis and prototype-based evaluation show constant-time online authentication per request, together with practical latency, throughput, and manageable membership-change overhead. Therefore, the proposed scheme provides a practical application-layer access-control architecture for AI-based enterprise document services.

Keywords: three-layer authentication protocol; secure access control; AI-based enterprise document management; session authentication; role/permission verification; group-membership authentication

Check for updates

Academic Editor: George Angelos Papadopoulos

Received: 17 September 2026
Revised: 27 October 2026
Accepted: 2 November 2026
Published: 6 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Enterprise document services are rapidly evolving into intelligent environments that combine Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), semantic retrieval, multimodal interfaces, and data analysis. In these systems, protected assets are no longer limited to stored documents, but also include derived objects such as chunks, embeddings, retrieval outputs, generated responses, and analytical results. This expansion makes secure service access more complex than in traditional document management systems.

Conventional access control mechanisms, including login-based authentication, Role-Based Access Control (RBAC), and Access Control Lists (ACLs)-based authorization, mainly focus on static user identity and document-level permission control [1–3]. However, in AI-based enterprise environments, secure access often depends not only on user identity and role, but also on session legitimacy and dynamic group conditions. In collaborative

settings, access decisions may vary according to project participation, workflow status, or event-dependent group membership.

Recent studies have addressed retrieval filtering, permission-aware RAG, and secure AI service operation [4–6]. In addition, AI-enabled service environments have motivated research on interpretable security mechanisms such as intrusion detection for industrial systems [7]. However, these studies mainly focus on filtering, anomaly detection, or monitoring, rather than on an integrated authentication architecture that verifies session validity, role legitimacy, and current group affiliation before service access.

To address this limitation, this paper proposes a three-layer authentication protocol for AI-based enterprise document management systems. The protocol sequentially performs session authentication, role/permission verification, and event-based group-membership authentication. In addition, event-specific group keys are used only as authentication material rather than as document re-encryption keys. This design supports dynamic access validation while preserving operational efficiency. The main contributions of this work are the design of the three-layer authentication structure, the event-based group authentication mechanism, the security analysis of the protected authentication values and group-key properties, and the prototype-based evaluation of latency, throughput, and membership-change overhead.

The remainder of this paper is organized as follows. Section 2 reviews related work on access control, group authentication, and security mechanisms for AI-based document service environments. Section 3 presents the preliminaries, notation, and security assumptions used throughout the paper. Section 4 describes the target system architecture and application environments. Section 5 defines the threat model and design goals of the proposed scheme. Section 6 presents the proposed three-layer authentication protocol in detail. Section 7 analyzes its security properties. Section 8 provides the efficiency analysis and implementation-based experimental evaluation. Section 9 discusses practical considerations and limitations. Finally, Section 10 concludes the paper.

2. Related Works

2.1. Access Control Models and System Authentication

Access control and system authentication have long been studied as fundamental components of information security systems, serving as the primary mechanisms for selectively granting users access only to authorized resources. Among them RBAC has been widely accepted as a representative access control model that structures permission management by introducing roles as intermediaries between users and permissions, thereby improving administrative efficiency and policy consistency in large-scale organizational environments [1]. Subsequently, access control research has expanded beyond RBAC to encompass a broader range of models and strategies, and recent studies have attempted to organize more systematically the conceptual definitions, taxonomies, and application conditions of authorization and access control [2,3].

The recent literature emphasizes that access control should not be viewed merely as a matter of permission assignment, but rather as a comprehensive security mechanism intertwined with system architecture, data characteristics, operational environments, and authentication procedures. Mohamed et al. systematically distinguished the definitions, strategies, and models of authorization and access control, and clarified the conceptual ambiguity and classification inconsistencies found in earlier studies [2]. In their subsequent work, they further compared the access control requirements and implementation characteristics of different database models, including relational, key-value, document-oriented, and graph databases, thereby demonstrating that the most suitable access control approach depends heavily on system structure and data organization [2]. Likewise, Farhadighalati

et al. showed through a recent systematic literature review that modern access control research is increasingly evolving toward the joint consideration of authentication, privacy, scalability, operational efficiency, and domain-specific requirements [3].

Meanwhile, in distributed and resource-constrained environments, the lightweightness and scalability of authentication procedures themselves have emerged as critical concerns. A comprehensive review of access control requirements and evaluation metrics in the Internet of Things (IoT) showed that, in real-world systems, access control and authentication must be evaluated together with computational cost, communication overhead, storage burden, and scalability [8]. In addition, a recent systematic review of IoT security emphasized that authentication protocols should not merely provide user identification, but should also remain implementable in practice while addressing diverse attack models and security requirements [9]. These developments provide important implications for Artificial Intelligence (AI)-based enterprise document management systems. In such environments, a single authentication mechanism based solely on login status is insufficient; rather, the legitimacy of the session, the appropriateness of user roles and permissions, and membership in a particular group or document set should be verified in a stepwise manner.

Against this background, this paper combines the policy-oriented structure of traditional access control models with the efficiency requirements emphasized in recent authentication research, and proposes a three-layer authentication architecture that sequentially performs session authentication, role/permission verification, and group-membership authentication. This approach reflects both the policy-level strengths of conventional access control models and the practicality and lightweightness emphasized in recent authentication protocol research [1–3,8,9].

2.2. Group Authentication and Authentication Protocols

Authentication and key management in group environments have long been recognized as central issues for secure group communication and dynamic membership control. Wong et al. proposed a key graph-based group key management scheme that structurally handles key update and distribution in large-scale group communication environments, and this work has since become one of the foundational references in group communication security and rekeying research [10]. Research in this line generally focuses on securely updating group keys and distributing them efficiently whenever group members join or leave.

More recently, lightweight authentication protocols and group authentication mechanisms have been actively investigated, particularly for dynamic and resource-constrained environments such as IoT. Alsheavi et al. conducted a comparative analysis of IoT authentication protocols and showed that existing approaches exhibit different strengths and weaknesses with respect to privacy protection, scalability, computational efficiency, and the heterogeneity of devices [11]. Shahidinejad and Abawajy provided a comprehensive analysis of blockchain-assisted authentication and session key generation protocols, highlighting not only the benefits of decentralized trust structures but also the increased protocol complexity and infrastructure overhead that accompany them [12]. Ansari and Ali reviewed lightweight cryptographic schemes and argued that hash-based or lightweight-operation-based approaches are advantageous in terms of computational efficiency, although trade-offs between security functionality and practical applicability remain inevitable depending on the target application scenario [13]. Furthermore, Ghaleb et al. systematically categorized lightweight authentication protocols for IoT communication networks and comparatively analyzed their computational cost, communication overhead, storage requirements, energy consumption, and security validation methods [14].

Research on group authentication protocols that directly support dynamic membership has also become more concrete in recent years. Feng et al. proposed a scalable group

authentication protocol based on PUF (Physical Unclonable Function)-derived bases, showing that both group authentication and key update can be performed with relatively low authentication latency and communication cost even in environments characterized by frequent member joining and leaving [15]. These recent studies collectively demonstrate that group authentication is no longer merely a problem of key sharing, but rather one that must simultaneously consider adaptability to membership changes, authentication latency, communication efficiency, and operational convenience.

Nevertheless, existing research on group authentication and authentication protocols has primarily focused on secure group communication at the network level, device authentication, session key agreement, or lightweight cryptographic mechanisms, rather than directly addressing an application-layer access-control architecture that integrates session state, role/permission validation, and group affiliation in service environments such as enterprise document management systems [10–15]. In particular, many existing group key-based approaches treat rekeying and redistribution themselves as the central issues under membership change. In contrast, the present paper utilizes the group key not simply as a communication protection key, but as event-based authentication material for group membership verification, and places it within a service access control structure linked with session and role verification. This constitutes a clear distinction from prior group authentication studies.

2.3. Access Control for RAG-Based Services

With the rapid expansion of Retrieval-Augmented Generation (RAG), access control for AI services that utilize internal or external knowledge resources during the retrieval stage has emerged as an important research topic. Since RAG-based services process user queries through multiple steps, including document retrieval, metadata filtering, context construction, and response generation, they require more fine-grained protection than traditional document-level access control. In enterprise environments, sensitive information may be indirectly exposed through retrieved document fragments or metadata, even when users do not directly access the original source documents themselves. Accordingly, access control at the retrieval stage has become essential [4,5].

In this context, Chen and Taipalus proposed a metadata-aware RAG architecture that combines metadata-based filtering with fine-grained access control, and demonstrated that both retrieval accuracy and access control can be strengthened simultaneously in sensitive data environments [4]. Jeong and Lee proposed a permission-aware RAG framework that directly interfaces with provider-controlled Identity and Access Management (IAM) systems in multi-resource environments, and showed that fine-grained access control can be enforced without redefining heterogeneous resource-specific policies into a single unified authorization model [5]. These studies suggest that, in RAG services, access control should not be treated as a simple post-processing filter, but rather should be integrated into the retrieval pipeline itself.

Furthermore, recent studies have begun to explore the use of RAG not merely as a supporting tool for static RBAC, but as a means of strengthening context-aware dynamic access decisions. Feng et al. proposed an access control framework for power systems that combines LLM and RAG on top of an RBAC-based access control structure, thereby enabling context-aware access decisions based on historical access logs and unstructured contextual information [6]. This line of work suggests that RAG may evolve beyond a retrieval aid into a more flexible decision-support mechanism for access control in dynamic operational environments.

However, current research on access control for RAG primarily focuses on retrieval-layer filtering or on strengthening context-aware authorization decisions, and does not

sufficiently address a multi-layer authentication structure that continuously verifies session legitimacy, role appropriateness, and group affiliation prior to service access [4–6]. In other words, while existing studies have made important progress in determining what should be included in retrieval results, they remain relatively limited in structuring who should be allowed to reach a given service stage and under what conditions at the application-service level.

In summary, prior studies on access control models have provided the theoretical foundations of policy structures and permission management [1–3,8], while research on authentication and group authentication has evolved around lightweightness, dynamic membership support, and rekeying efficiency [9–15]. More recently, studies on RAG-based services have highlighted the need for fine-grained access control at the retrieval stage [4–6]. However, these three research streams have largely developed independently. As a result, there remains insufficient research that integrates session-level legitimacy verification, policy-based role confirmation, and event-based group membership verification into a single coherent authentication structure for AI-based enterprise document management environments. To bridge this gap, this paper proposes a three-layer authentication protocol that combines conventional access control models, group authentication mechanisms, and the access control requirements of RAG-based services.

3. Preliminaries

3.1. Security Building Blocks

This section defines the fundamental cryptographic components required to describe the proposed three-step authentication protocol.

Definition 1. ***Pseudo-Random Permutation (PRP).***

Let K be a key space, and let $\{0,1\}^l$ denote the space of bit strings of length l. A family of functions $\mathcal{F} = \{f_k\}_{k\in K}$ is called a Pseudo-Random Permutation (PRP) if it satisfies the following conditions:

1. *For every $k \in K$, f_k: $\{0,1\}^l \to \{0,1\}^l$ is a bijection and is efficiently invertible.*
2. *For any probabilistic polynomial-time adversary, $f_k(\cdot)$ is computationally indistinguishable from a uniformly chosen random permutation over the same domain and range.*

In this paper, such a PRP is denoted by f, where the forward operation is written as $f_k(\cdot)$ and the corresponding inverse operation is written as $f_k^{-1}(\cdot)$ [16].

Definition 2. ***Key-Derivation Function (KDF).***

A Key-Derivation Function (KDF) is a function that takes as input a secret value X and a public context string ctx, and outputs a derived key of fixed length. In this paper, it is denoted by

$$KDF\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^l.$$

Different context strings are used to distinguish keys for different purposes and to ensure domain separation [16,17].

Notation and Encoding Convention.

In this paper, σt, St, and Rt are all assumed to be encoded as fixed-length bit strings of length l. Accordingly, the XOR operation among these values is performed only on bit strings of equal length. In addition, ctx_2 and ctx_3 denote fixed public context strings associated with Layer 2 and Layer 3, respectively. These strings are used for domain separation in order to distinguish the purposes of cryptographic operations performed at different layers [16,17]. In addition, one notation convention is applied to all hash-chain variables, both in formulas and in body text: the group index is always written as a subscript and the event index

as a superscript (e.g., α_i^t and β_i^t), and a hash function applied k times is written as $H^k(\cdot)$ with explicit parentheses. No alternative ordering of these indices, and no parentheses-free application notation, is used anywhere in this paper.

Definition 3. ***Security Requirements of Group Keys***

The proposed protocol considers the following security requirements for event-based group keys.

(1) ***Group Key Secrecy*** *means that an adversary cannot distinguish the valid group key of a target event from a uniformly random bit string of the same length.*

(2) ***Forward Secrecy*** *means that a user who is removed from a group, or an adversary who learns the current event-related authentication material, cannot derive any future event group key.*

(3) ***Backward Secrecy*** *means that a newly joined user, or an adversary who learns the current event-related authentication material, cannot derive any past event group key [18].*

These three requirements are essential in dynamic group environments where membership changes may occur over time.

Definition 4. ***One-Wayness of Hash Function***

A hash function H: $\{0,1\}^* \rightarrow \{0,1\}^n$ is said to satisfy one-wayness if, for a given output value $y = H(x)$, it is computationally infeasible for any PPT adversary to recover a valid preimage x with non-negligible probability [19,20].

Definition 5. ***Properties of Hash Chain***

A hash chain is constructed by repeatedly applying a one-way hash function to an initial seed value. A sequence defined as $x_i = H(x_{i-1})$ for an initial secret x_0 and a hash function H is called a hash chain. In this paper, two complementary hash-chain directions are used in the event-based group-key generation process.

1. *A normal hash chain is used to generate the sequence, where each subsequent value is obtained by forward application of the hash function. Due to the one-wayness of H, the current value does not reveal its predecessor, which supports backward secrecy, while subsequent values remain efficiently computable from the current one, providing forward accessibility.*
2. *A reversed hash chain is used to generate the sequence, where the event ordering is defined in the opposite direction. Under the same one-wayness assumption, the current value does not reveal the value of a later event in the reversed direction, thereby supporting forward secrecy, while values corresponding to earlier positions in the reversed chain remain derivable, providing backward accessibility [19,20].*

Based on these building blocks, the proposed protocol constructs the event-specific group key for group Gi at event t as $GiK_t = f_{gi}(\alpha_i^t + \beta_i^t)$, where gi is the master secret of group Gi, α_i^t is derived from a normal hash chain, and β_i^t is derived from a reversed hash chain. This combined structure is designed so that compromise of the current event-related material does not directly enable derivation of either past or future group keys, while still allowing efficient event-wise key generation.

3.2. Security Games

Definition 6. ***Security Game ICR-3layAP (Indistinguishability of Ciphertexts from Random Bit Strings in 3layAP (3-Layer Authentication Protocol).***

This game is a modification of the ***Chosen Plaintext Attack (CPA)*** *model. The purpose of this game is to show that, even if an adversary observes legitimate protocol executions and performs valid queries, the ciphertexts generated during the authentication process remain computationally indistinguishable from uniformly random bit strings of the same length.*

Setup. Given a security parameter λ, the challenger C generates the public system parameters. This process includes the selection of a hash function H, a Pseudo-Random

Permutation (PRP) f, a user set, a role set, and a group set, as well as the generation of each user's secret key km, each group's master key gi, and the event-specific group key GiK_t. In addition, the challenger initializes the user registration information and the public parameters of the system [18,21].

Queries. The adversary A may adaptively issue the following queries:

1. Registration Query: Requests user registration information.
2. Login Query: Requests the session-authentication ciphertext generated upon a successful login, defined as $C_{m,t}^{(1)} = f_{km}(St \parallel \sigma t)$.
3. Role-Verification Query: Requests the role-authentication ciphertext generated upon successful role/permission verification, defined as $C_{m,t}^{(2)} = f_{\sigma t}(St \oplus Rt)$.
4. Group-Authentication Query: Requests the group-authentication ciphertext generated upon successful group-membership authentication, defined as $C_{i,t}^{(3)} = f_{GiK_t t}(G_i_k_t)$.

Challenge. The adversary A selects an authentication stage $\ell \in \{1, 2, 3\}$ as the challenge target, together with the corresponding user, session, role, or group. Here, $\ell = 1$ denotes the login and session-issuance stage, $\ell = 2$ denotes the role/permission verification stage, and $\ell = 3$ denotes the group-membership authentication stage. The challenger C then chooses a random bit $b \in \{0, 1\}$. If $b = 1$, the challenger returns the real authentication ciphertext C^*; otherwise, if $b = 0$, it returns a uniformly random bit string R^* of the same length.

Guess. The adversary A outputs a guess bit b', indicating whether the received value is a real ciphertext or a random bit string. The adversary's advantage is defined as

$$Adv_A^{ICR-3layAP}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

If the above advantage is negligible for any Probabilistic Polynomial-Time (PPT) adversary A, then the authentication-related ciphertexts of the proposed protocol are said to be secure in the sense of ICR-3layAP.

3.3. Notations

Um	The m-th user
SA	System Administrator
km	Long-term secret key of user Um
Rj	The j-th role
Gi	The i-th group
gi	Master key of group Gi
GiK_t	Event-specific group key of group Gi at event t
St	Session identifier at time t randomly generated every session
σt	Session authentication token randomly generated every session
Rt	Role authentication token randomly generated every session
$G_i_k_t$	Group authentication token randomly generated every session
Dk	The k-th protected document or data object
α_i^t	The hashed values from α_i^0 for group Gi in each event t
β_i^t	The reverse hashed values from β_i^0 for group Gi in each event t
$f_k()$	The pseudo-random permutation with encryption key k
$H()/H^q()$	Hash function/q-times hash function

4. Application Environments and System Configuration

4.1. Application Environment

It is assumed that a web-based enterprise service that integrates multilingual economic data analysis and intelligent document management. Users can upload, view, download, and query corporate documents, while the system also supports financial spreadsheet analysis, external economic data integration, image-based question answering, Speech-to-Text (STT), and Text-to-Speech (TTS). Such an environment is consistent with recent

RAG-based enterprise services, in which access control must be applied not only to stored documents but also to retrieval and response generation [4,22,23].

4.2. Overall System Configuration

The system consists of a user interface layer, an authentication and authorization layer, an encrypted storage layer, a question-answering layer, an economic analysis layer, and an administration layer. We assume such a general Artificial Intelligence (AI)-based enterprise document management architecture and introduce an additional 3-layer authentication protocol into it. In the proposed design, the group key is used for group membership verification and authentication token issuance, rather than for document re-encryption [18].

Figure 1 illustrates the overall access-control architecture considered in this work. A user request first enters the authentication and authorization layer, where the system performs login and session issuance, role/permission verification, and event-aware group-membership authentication in sequence. Only after these three layers are successfully completed can the user access the protected service modules, including encrypted storage, document question answering, RAG-based and multimodal processing, and economic data analysis. The administration module manages users, roles, groups, permissions, and document-access policies, while the lower part of the figure emphasizes that the protected assets include not only stored documents but also intermediate and derived artifacts produced during retrieval, generation, and analysis.

4.3. Authentication and Authorization Management Module

This module performs login, session management, role verification, and permission verification. After account authentication, users undergo RBAC- or ACL-based checking, and protected organizational resources require an additional group-based authentication step. All subsequent retrieval, generation, and analysis functions depend on the results of this module [18,22].

4.4. File Upload and Encrypted Storage Module

This module stores uploaded documents after encrypting original files with Advanced Encryption Standard-256 in Galois/Counter Mode (AES-256-GCM), while metadata are managed separately. The storage layer protects document contents, whereas the authentication layer verifies user eligibility. Accordingly, the group key is not used for document re-encryption, but only for session-level group membership verification and token issuance.

4.5. Document Question-Answering Module

This module extracts text from PDF, Document (DOCX), Microsoft Excel Open XML Spreadsheet (XLSX), and CSV (Comma-Separated Values) files, divides the text into chunks, and generates embeddings. These chunks and embeddings constitute the preprocessing stage of the RAG pipeline and are directly used for subsequent retrieval and response generation [4]. Therefore, they must also be treated as protected resources.

4.6. RAG-Based Question-Answering and Multimodal Processing Module

The RAG-based question-answering and multimodal processing module combines retrieval and generation functions. For a user query, the system performs embedding-based retrieval and provides relevant document chunks to the response generation model in order to produce evidence-grounded answers. It also supports image question answering, Speech-to-Text (STT), and Text-to-Speech (TTS), thereby processing non-textual input and output as well.

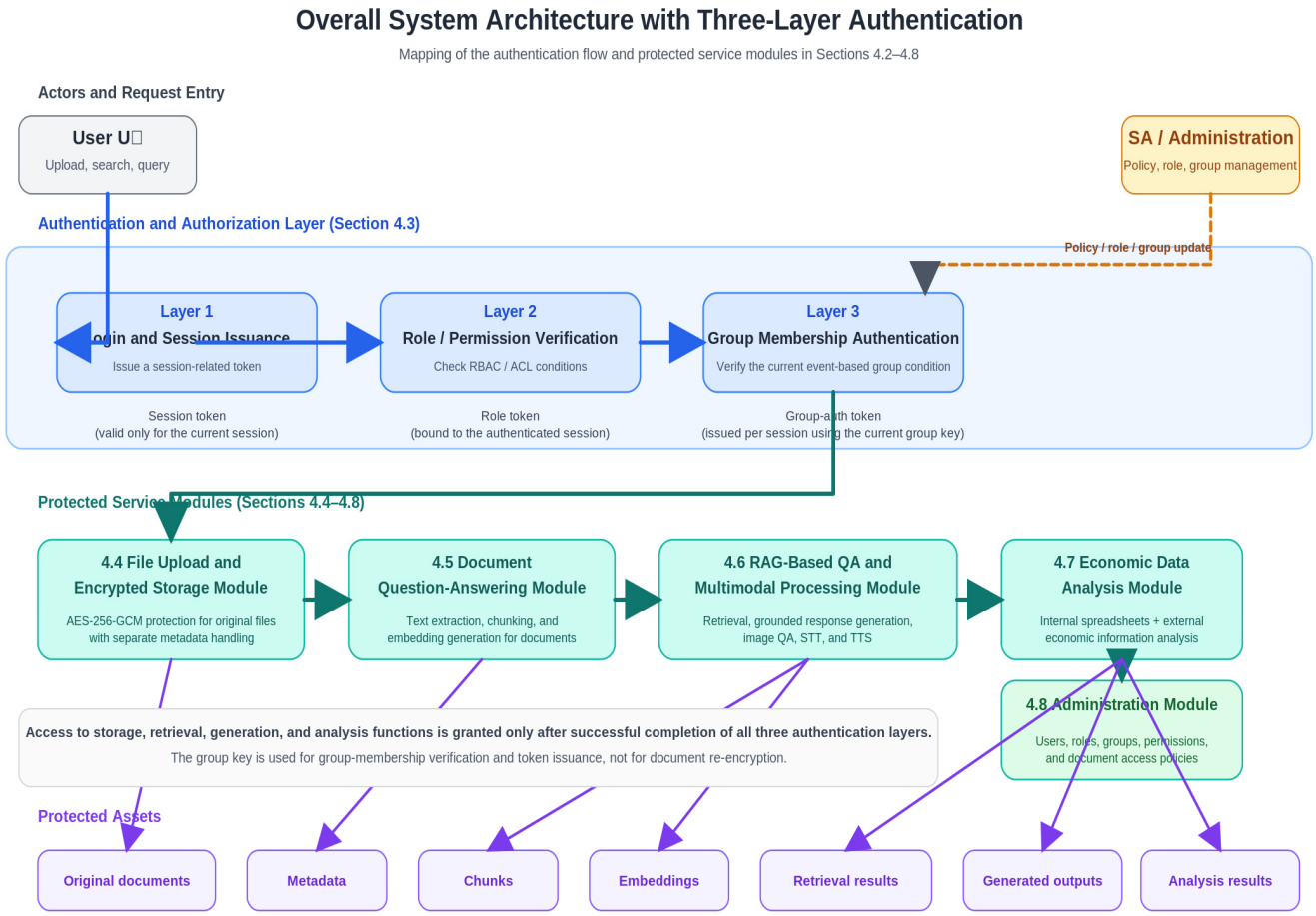


Figure 1. Overall system architecture of the proposed AI-based enterprise document management environment. The figure shows how the three-layer authentication procedure—login and session issuance, role/permission verification, and group-membership authentication—controls access to the protected service modules described in Sections 4.3–4.8. The protected objects include not only original documents and metadata, but also chunks, embeddings, retrieval results, generated outputs, and analysis results. Annotation. The three verification layers map onto the modules of Section 4 as follows. Layer 1 (login and session issuance) is executed in the authentication and authorization management module (Section 4.3). Layer 2 (role/permission verification) is executed in the same module, using the role and resource policies administered in Section 4.8. Layer 3 (group-membership authentication) gates the group-protected service modules, namely encrypted storage (Section 4.4), document question answering (Section 4.5), RAG-based and multimodal processing (Section 4.6), and economic data analysis (Section 4.7). Every request traverses Layer 1 first, and the session identifier issued there is the value against which Layers 2 and 3 are bound.

This module includes not only original documents but also retrieval results, response contexts, and generated outputs among the protected objects. Therefore, authentication and authorization verification should not be applied only at the document storage stage, but must also be extended to the retrieval and generation stages [4,22].

4.7. Economic Data Analysis Module

This module combines internal financial spreadsheets with external economic information to derive indicators related to profitability, growth, and capital structure. Because its outputs may directly support managerial decision-making, they must also be treated as protected resources subject to authentication and authorization.

4.8. Administration Module

This module manages users, roles, groups, permissions, and document-access policies. Since role or group changes directly affect subsequent access decisions, the administration function is essential for maintaining the correctness of authentication and access control [18,22].

The application environment considered in this paper integrates document storage, semantic retrieval, RAG-based response generation, economic data analysis, and multimodal interaction within a web-based enterprise document management system. In this environment, protected objects include not only original documents but also chunks, embeddings, retrieval results, generated responses, and analytical outputs. Therefore, service access control must consider not only account authentication and policy verification but also role and group membership within the current session.

5. Threat Model

This section defines the potential security threats considered in the Artificial Intelligence (AI)-based enterprise document management environment to which the proposed protocol is applied. The system integrates document storage, retrieval, RAG-based question answering, economic data analysis, and multimodal services, while access control is enforced through account authentication, role verification, and group membership verification. Accordingly, the threat model must cover not only unauthorized access by external users, but also misuse and collusion among legitimate users that may arise in group-based collaborative environments.

In this paper, the adversary is assumed to know the public system architecture and protocol procedures. In addition, the adversary may attempt to eavesdrop on, replay, or modify messages transmitted over the network, or abuse partial authentication information obtained from a legitimate user account or an active session. Therefore, the threat model focuses on attack scenarios that may realistically occur in the application-layer authentication process.

5.1. Adversarial Model

The adversary is divided into an external adversary and an internal adversary. An external adversary is a user who is not registered in the system or does not possess a valid session token, role authentication token, or group authentication token. An internal adversary is a legitimate registered user who attempts to access roles, groups, or resources that have not been assigned to that user. This paper particularly considers scenarios in which internal and external adversaries are combined in a group environment. This is because, in group-based systems, the object of authentication is not limited to account identity or role appropriateness, but also includes legitimate group membership at the current point in time [4,18].

5.2. Protected Assets

The protected assets considered in this paper are as follows. First, the original documents and metadata in the storage layer must be protected. Second, the chunks and embeddings generated during document processing must also be protected. Third, application-layer outputs, such as retrieval results, response contexts, generated responses, and financial analysis results, are treated as protected assets. Fourth, the session authentication token σt, the role authentication token Rt, the group authentication token $G_i_k_t$, and the event-specific group key GiK_t used in the authentication process are protected. In particular, the last category is regarded as a core protected asset because it directly determines group-based access control [4,22].

5.3. Threat Scenarios

5.3.1. Impersonation and Unauthorized Access

The first threat arises when an adversary impersonates a legitimate user or attempts to access roles, groups, or data beyond its authorization scope. An external adversary may attempt to bypass the session authentication procedure by forging login requests or authentication messages, while an internal adversary may attempt to access resources outside its current role or group scope. Therefore, the system requires not only account authentication, but also session-bound role verification and group verification [4,22].

5.3.2. Replay and Token Misuse

The second threat arises when authentication-related ciphertexts or tokens obtained from a previous session or event are reused in the current session. An adversary may attempt to bypass authentication by replaying a session authentication token, a role authentication token, or a group authentication token. In addition, an adversary may attempt to reuse authentication information obtained at one stage without undergoing verification at another stage. Therefore, each authentication token must be bound to a specific session or event, and authentication information belonging to different layers must be verified independently.

Accordingly, previously observed or replayed authentication values cannot be reused to satisfy the verification requirements of a new session. The proposed protocol focuses on application-layer authentication and access control. Integrity and authenticity of transmitted messages are assumed to be provided by a standard secure communication channel, such as TLS, used together with the proposed protocol.

5.3.3. Collusion in Group Environments

The third threat is collusion in a group environment. An adversary may collude with a user who is a legitimate group member in the current event and may receive from that user the group authentication material or part of the group key valid at that time. The adversary may then attempt to use such information to perform operations for a group to which it does not belong or to access unauthorized group resources. This paper regards such a threat as a key attack in group environments, because, in group-based collaborative systems, even if account authentication and role verification succeed, protection of fine-grained group resources cannot be guaranteed unless legitimate group membership at the current point in time is separately verified [18].

5.3.4. Inter-Event Key Derivation

The fourth threat arises when an adversary attempts to derive the group key or authentication material of a previous or subsequent event from the group authentication material obtained in a specific event. If an adversary can compute GiK_{t-1} or GiK_{t+1} from the current event group key GiK_t or related authentication information, then a newly joined user may gain access to past-event resources, or a withdrawn user may continue to access subsequent-event resources. Therefore, resistance to inter-event key derivation is an essential requirement for maintaining the validity of group-based access control [18–20].

5.4. Security Objectives

From the above threat model, the security objectives can be summarized as follows. First, an unauthorized user must not be able to forge a session authentication token, role authentication token, or group authentication token, nor generate a value that is distinguishable from a valid one. Second, a user must not be able to access resources beyond its current role or group scope. Third, authentication values from previous sessions or previous events must not be reusable at the current time. Fourth, valid group keys for

previous or subsequent events must not be computable solely from the current event's group authentication material. Fifth, even through collusion with a legitimate user, an adversary must not be able to obtain authentication information for a group or event that is not authorized for that adversary.

6. Three-Layer Authentication Protocol

The considered environment consists of a trusted management entity and organizational users. Specifically, SA denotes the System Administrator, which is assumed to act as a Trusted Third Party (TTP) for system initialization, credential management, and security parameter administration. In contrast, Um denotes the m-th user in the system, namely, a member belonging to an organizational unit or access group within the enterprise document environment. Thus, the proposed protocol is designed for interactions between a trusted administrative authority and authorized organizational members who request access to protected resources.

6.1. Preparation Phase

This process is carried out by the SA.

6.1.1. Generation of Group Keys and Secret Values Required at Each Stage

(1) Group-Key Generation for Group i in Each Event:

The first event group key for the i-th group Gi: $GiK^1 = f_{gi}(\alpha_i^1 + \beta_i^1)$
The second event group key for the i-th group Gi: $GiK^2 = f_{gi}(\alpha_i^2 + \beta_i^2)$
$\vdots$
The q-th event group key for the i-th group Gi: $GiK^q = f_{gi}(\alpha_i^q + \beta_i^q)$
(Explanation). GiK^q: the q-th event group key for group Gi
gi is the master key for group Gi
$f_{gi}()$ is the pseudo-random function with encryption key gi
α_i^t, β_i^t are secret values for group Gi in each event t
* Generation of each α_i^t and β_i^t
α_i^0 and β_i^0 are selcected randomly. Then, they are hashed q-times repeatedly.
$H(\alpha_i^0) = \alpha_i^1$, $H(\beta_i^0) = \beta_i^q$. The secret values of α_i^t and β_i^t from the first event to the q-th event are as shown in the following table.

event t	hashed value α_i^t	hashed value β_i^t
1	$H(\alpha_i^0) = \alpha_i^1$	$\beta_i^1 = H^q(\beta_i^0)$
2	$H(\alpha_i^1) = H(H(\alpha_i^0)) = H^2(\alpha_i^0) = \alpha_i^2$	$\beta_i^2 = H^{q-1}(\beta_i^0)$
3	$H(\alpha_i^2) = H(H^2(\alpha_i^0)) = H^3(\alpha_i^0) = \alpha_i^3$	$\beta_i^3 = H^{q-2}(\beta_i^0)$
$\vdots$	$\vdots$	$\vdots$
t	$H(\alpha_i^{t-1}) = H(H^{t-1}(\alpha_i^0)) = H^t(\alpha_i^0) = \alpha_i^t$	$\beta_i^t = H(\beta_i^{t+1}) = H(H^{q-t}(\beta_i^0)) = H^{q-t+1}(\beta_i^0)$
$\vdots$	$\vdots$	$\vdots$
$q-1$	$H(\alpha_i^{q-2}) = H(H^{q-2}(\alpha_i^0)) = H^{q-1}(\alpha_i^0) = \alpha_i^{q-1}$	$\beta_i^{q-1} = H(\beta_i^q) = H(H(\beta_i^0)) = H^2(\beta_i^0)$
q	$H(\alpha_i^{q-1}) = H(H^{q-1}(\alpha_i^0)) = H^q(\alpha_i^0) = \alpha_i^q$	$\beta_i^q = H(\beta_i^0)$

As shown in the table, the α_i^t value becomes the secret value corresponding to each event in the order of the number of times the initial value is hashed. Conversely, for β_i^t, hashing the initial value once results in the last q-th event value, hashing it one more time results in the $(q-1)$-th event value, and hashing it q times results in the first event value. In other words, α_i^0 uses q hash chains in order as the secret values for the events, while β_i^0 uses them in reverse order.

(2) Generation of Other Secret Values Required at Each Stage: secret keys $\{km\}$ for each user, roles $\{Rj\}$, group number $\{i\}$ and the group secret values $\{\alpha_i^t, \beta_i^t\}$ for group $\{i\}$ and t-th event group key $\{GiK^t\}$.

6.1.2. User Registration

Um —-> System Administrator (SA): IDm||PWm.
SA stores: IDm||H(PWm).
Um <—- SA: secret key km, role Rj, group number i and t-th event group key GiK_t.
(Explanation) In the registration and login phases, user Um enters the password PWm, and the system processes it only in hashed form as H(PWm). Thus, the plaintext password is neither stored nor revealed to the system administrator SA, which helps reduce the risk of insider misuse. This treatment is also consistent with Korean personal-information protection requirements, under which passwords should be stored in a one-way encrypted form rather than in recoverable plaintext [24]. Here, H(PWm) denotes a password-derived verifier used by the authentication protocol. Practical password-hardening mechanisms are assumed to be provided by a companion credential-management mechanism and are outside the scope of this protocol. These registered values are subsequently used in the three-layer authentication procedure.

6.2. Actual Use Phase

6.2.1. Login and Session Issuance

$$\text{Um —-> SA: IDm||H(PWm)} \tag{1}$$

SA verifies: IDm||H(PWm)

If successful,

$$\text{SA issues to Um: } (f_{km}(St\|\sigma t) \tag{2}$$

$$\text{Um decrypts: } f^{-1}(f_{km}(St\|\sigma t)) = (St\|\sigma t) \tag{3}$$

(Explanation). St means the t-th session, σt is the authentication token for the session St.

In this process, if user authentication is successful using an ID and password, the system issues an authentication token for that session to the user Um, encrypted with the user's private key km.

6.2.2. Policy-Based Role/Permission Verification

$$\text{Um queries: } IDm\|f_{\sigma t}(St\|Rj\|I\text{-}Dk) \tag{4}$$

SA decrypts and verifies: St and Rj

If successful,

$$\text{System sends to Um: } f_{\sigma t}(ctx_2\|St\oplus Rt), f_{Rt}(Dk) \tag{5}$$

$$\text{Um decrypts: } f^{-1}(f_{\sigma t}(ctx_2 \parallel St \oplus Rt)) = ctx_2 \parallel (St \oplus Rt) \tag{6}$$

After confirming the expected prefix ctx_2,

$$\text{computes: } (St\oplus Rt)\oplus St = Rt \tag{7}$$

$$\text{decrypts: } f^{-1}(f_{Rt}(Dk)) = Dk \tag{8}$$

(Explanation). $I\text{-}Dk$: data Dk's index; Rt: authentication token for Um's Rj in the t-th session.

When Um requests data *Dk* encrypted with the session token σt, the SA decrypts it using σt, which is mapped to IDm and the current session, to verify the current session St and Um's role *Rj*. If verification is successful, the authentication token *Rt*, confirming the successful *Rj* verification, is encrypted with the session token σt and transmitted, and the requested *Dk* is encrypted with the role verification token *Rt* and transmitted (5). Here, *St* and *Rt* are encoded as fixed-length bit strings of the same length *l*.

In (6), the protected Layer 2 value is interpreted as $ctx_2 \| (St \oplus Rt)$, where ctx_2 is a fixed public context string of length d for Layer 2. After decryption, Um first verifies the expected prefix ctx_2 and the predefined total length, and then interprets the remaining *l*-bit substring as $St \oplus Rt$. Since *St* and *Rt* are encoded as equal-length bit strings, Um obtains *Rt* by computing $(St \oplus Rt) \oplus St = Rt$. Any value that does not match the expected prefix or fixed-length format is rejected.

6.2.3. Group-Membership Authentication

If the user has passed the previous verification processes and group-level detailed task execution is required, the group-key-based authentication procedure is performed.

$$\text{Um queries: } IDm \| f_{K_t^{(3)}}\left(i \| GiK^t\right) \tag{9}$$

$$\text{here, } K_t^{(3)} = KDF(kt \| ctx_3), kt = \sigma t \oplus Rt \tag{10}$$

$$\text{SA decrypts and verifies: } f^{-1}\left(f_{K_t^{(3)}}\left(i \| GiK^t\right)\right) = \left(i \| GiK^t\right) \tag{11}$$

If successful,

$$\text{System issues: } \left(f_{GiK^t}(G_i_k_t)\right) \tag{12}$$

$$\text{Um decrypts: } f^{-1}\left(f_{GiK^t}(G_i_k_t)\right) = G_i_k_t \tag{13}$$

(Explanation). In the current session St, after obtaining σt and *Rt* mapped to *IDm*, the system first computes the intermediate value $kt = \sigma t \oplus Rt$. Instead of directly using *kt* as the Layer 3 cryptographic key, this protocol derives the actual Layer 3 key as $K_t^{(3)} = KDF(kt \| ctx_3)$, where ctx_3 is the fixed public context string of length d for Layer 3. The KDF is used to generate a fixed-length key for group authentication and to provide explicit domain separation. The verification is successful, using this derived key, SA issues a group verification token *G_i_k_t*, which means authentication token for session *t* at the *k*-th event of group *i*.

In that session, thereafter, Um uses the access-authorization token for group data, namely *G_i_k_t*, as the encryption key to continue group-related operations (request/response).

6.3. Whole Protocol

User *Um*	*SA* (*System* Administrator)
1. Preparation phase	
1.1. Generation of Group Keys and Secret Values Required at Each Stage (1) Group-Key Generation for Group *i* in Each Event: $GiK^t = f_{gi}(\alpha_i^t + \beta_i^t)$ $= f_{gi}\left(H^t(\alpha_i^0) + H^{q-t+1}(\beta_i^0)\right)$ (2) Generation of Other Secret Values Required at Each Stage: {km}, {Rj}, group number {i} and $\{\alpha_i^t, \beta_i^t\}$	
1.2. User Registration $\xrightarrow{\text{IDm}\|\text{PWm}}$	Store: IDm ‖ H(PWm) $\xleftarrow{km, \text{ role } Rj, \text{ group number } i, GiK^t}$

2. Actual use phase	
2.1. Login and Session Issuance Input: $\xrightarrow{\text{IDm}\|\text{H(PWm)}}$ Decrypt: $f^{-1}(f_{km}(St \parallel \sigma t)) = (St \parallel \sigma t)$	Verify: IDm ‖ H(PWm) $\xleftarrow{f_{km}(St\|\sigma t)}$
2.2. Policy-Based Role/Permission Verification Query: $\xrightarrow{IDm\|f_{\sigma t}(St\|Rj\|I-Dk)}$ Decrypt & Compute: $f_{\sigma t}^{-1}(f(ctx_2 \parallel St \oplus Rt)) = ctx_2 \parallel St \oplus Rt$ $(St \oplus Rt) \oplus St = Rt, \quad f_{Rt}^{-1}(f(Dk)) = Dk$	Verify: St and Rj $\xleftarrow{f_{\sigma t}(ctx_2\|St\oplus Rt),\, f_{Rt}(Dk)}$
2.3. Group-Membership Authentication Query: $\xrightarrow{IDm\|f_{K_t^{(3)}}(i\|GiK^t)}$ Decrypt: $f_{GiK^t}^{-1}(f(G_i_k_t)) = G_i_k_t$	Decrypt and Verify: $f^{-1}(f_{K_t^{(3)}}(i \parallel GiK^t)) = (i \parallel GiK^t)$ Issue: $\xleftarrow{f_{GiK^t}(G_i_k_t)}$

Resource-Level Policies (Document-Level Policy Clarification)

In the proposed framework, access control is not limited to coarse-grained user login or static role assignment. Each protected resource may be associated with a policy context that includes the validated session state, the authorized role or permission set, and the current group-event condition. Accordingly, the same three-layer procedure can be applied not only to original documents, but also to derived objects such as document chunks, embeddings, retrieval results, and analytical outputs. This enables resource-level protection in which different documents or derived resources may be governed by different authorization conditions even within the same organizational group.

7. Security Analysis

7.1. Security Properties of Event-Based Group Keys

In the proposed protocol, the event-specific group key for group Gi at event t is defined as $GiK^t = f_{gi}(\alpha_i^t + \beta_i^t)$, where f is a pseudo-random permutation, gi is the master key of group Gi, α_i^t is generated from a normal hash chain, and β_i^t is generated from a reversed hash chain. More specifically, $\alpha_i^t = H^t(\alpha_i^0)$ and $\beta_i^t = H^{q-t+1}(\beta_i^0)$ for initial seeds α_i^0 and β_i^0, and for an event index $t \in \{1, \ldots, q\}$.

Throughout this section, we assume the following.

Assumption 1-1. *f is a secure pseudo-random permutation (PRP) [16].*

Assumption 1-2. *f is a one-way hash function [19,20].*

Assumption 1-3. *The master key gi and the initial seeds α_i^0, β_i^0 are not exposed to the adversary.*

Operational note on master-key lifecycle. Assumption 1-3 states that the group master key gi and the initial seeds are not exposed to the adversary. If gi is compromised or requires periodic rotation, the affected group should be re-initialized with a fresh master key and fresh initial seeds, and the corresponding authentication materials should be regenerated and redistributed. Accordingly, master-key rotation and compromise recovery are treated as operational key-management procedures outside the online authentication path, consistent with NIST SP 800-57 [25].

We now show that the proposed event-based group key construction satisfies group key secrecy, forward secrecy, and backward secrecy.

Theorem 1. ***Group Key Secrecy***

For any probabilistic polynomial-time adversary A, the event-specific group key GiK_t is computationally indistinguishable from a uniformly random bit string of the same length.

Proof. Assume, toward contradiction, that there exists a PPT adversary A that can distinguish GiK^t from a random bit string with non-negligible advantage. We construct a PPT distinguisher B that breaks the PRP security of f.

Let B be given oracle access to O, where O is either the real PRP $f_{gi}(\cdot)$ or a truly random function with the same domain and range. The distinguisher B honestly chooses α_i^0 and β_i^0, computes α_i^t and β_i^t, and defines $x_t = \alpha_i^t + \beta_i^t$.

It then queries the oracle on x_t and obtains $Y^* = O(x_t)$.

The value Y^* is given to A as the challenge key.

If $O = f_{gi}$, then Y^* is distributed exactly as the real group key GiK_t. If O is a truly random function, then Y^* is distributed as a uniformly random string. Therefore, any non-negligible distinguishing advantage of A directly yields a non-negligible distinguishing advantage for B, contradicting Assumption 1-1.

Hence, GiK^t is computationally indistinguishable from a random bit string, and the proposed construction satisfies group key secrecy. □

Theorem 2. ***Forward Secrecy***

Let $Adv^{FS}_{3layAP}(A, \lambda)$ denote the advantage of a probabilistic polynomial-time adversary A in distinguishing the real future event-specific group key from a uniformly random bit string of the same length after learning the current event-related material at event t, namely α_i^t, β_i^t, and GiK_t. Then, for any probabilistic polynomial-time adversary A, there exist probabilistic polynomial-time adversaries B1 and B2 such that

$$Adv^{FS}_{3layAP}(A,\lambda) \le Adv^{\mathrm{OWF-H}}(\mathrm{B1},\lambda) + Adv^{\mathrm{PRP-f}}(\mathrm{B2},\lambda) + negl(\lambda).$$

Proof. Consider a future event t′ > t. In the proposed construction, the corresponding future event-specific group key is computed as $GiK^{t'} = f_{gi}(\alpha_i^{t'} + \beta_i^{t'})$.

We define two hybrid games. Game G0 is the real experiment, in which the challenger returns the real future group key $GiK^{t'}$. Game G1 is identical to G0, except that the challenger replaces $GiK^{t'}$ with a uniformly random bit string R^* of the same length.

If adversary A can distinguish G0 from G1 with non-negligible advantage, then one can construct a probabilistic polynomial-time adversary B2 that distinguishes the pseudo-random permutation f from a truly random string with the same non-negligible advantage. Therefore,

$$|Pr[\text{A wins in G0}] - Pr[\text{A wins in G1}]| \le Adv^{\mathrm{PRP-f}}(\mathrm{B2},\lambda).$$

Next, consider G1. Since the challenge value in G1 is independent of the real future group key, adversary A can obtain a non-negligible advantage only if it can recover the hidden future input required for the genuine computation of $GiK^{t'}$. From the currently known α_i^t, the adversary may derive $\alpha_i^{t'}$ by repeated forward application of H. However, computing $GiK^{t'}$ also requires $\beta_i^{t'}$. Since $\beta_i^{t'}$ lies in the preimage direction relative to the currently known β_i^t along the reverse hash chain, deriving $\beta_i^{t'}$ from β_i^t would contradict the one-wayness of H. Hence, one can construct a probabilistic polynomial-time adversary B1 such that

$$|Pr[\text{A wins in G1}] - 1/2| \le Adv^{\mathrm{OWF-H}}(\mathrm{B1},\lambda) + \mathrm{negl}(\lambda).$$

Combining the above inequalities, we obtain the stated bound. Therefore, the proposed construction satisfies forward secrecy. □

Theorem 3. ***Backward Secrecy***

Let $Adv^{BS}_{3layAP}(A, \lambda)$ *denote the advantage of a probabilistic polynomial-time adversary A in distinguishing the real past event-specific group key from a uniformly random bit string of the same length after learning the current event-related material at event t, namely* α_i^t, β_i^t, *and* GiK^t. *Then, for any probabilistic polynomial-time adversary A, there exist probabilistic polynomial-time adversaries C1 and C2 such that*

$$Adv^{BS}_{3layAP}(A,\lambda) \leq Adv^{\mathrm{OWF-H}}(\mathrm{C1},\lambda) + Adv^{\mathrm{PRP-f}}(\mathrm{C2},\lambda) + negl(\lambda).$$

Proof. Consider a past event t′ < t. In the proposed construction, the corresponding past event-specific group key is computed as $GiK^{t'} = f_{gi}(\alpha_i^{t'} + \beta_i^{t'})$.

We again define two hybrid games. Game G0 is the real experiment, in which the challenger returns the real past group key $GiK^{t'}$. Game G1 is identical to G0, except that the challenger replaces $GiK^{t'}$ with a uniformly random bit string R^* of the same length.

If adversary A can distinguish G0 from G1 with non-negligible advantage, then one can construct a probabilistic polynomial-time adversary C2 that distinguishes the pseudo-random permutation f from a truly random string with the same non-negligible advantage. Therefore,

$$|Pr[\text{A wins in G0}] - Pr[\text{A wins in G1}]| \leq Adv^{\mathrm{PRP-f}}(\mathrm{C2},\lambda).$$

Next, consider G1. Since the challenge value in G1 is independent of the real past group key, adversary A can obtain a non-negligible advantage only if it can recover the hidden past input required for the genuine computation of $GiK^{t'}$. From the currently known β_i^t, the adversary may derive $\beta_i^{t'}$ by repeated forward application of H along the reverse hash chain. However, computing $GiK^{t'}$ also requires $\alpha_i^{t'}$. Since $\alpha_i^{t'}$ lies in the preimage direction relative to the currently known α_i^t along the forward hash chain, deriving $\alpha_i^{t'}$ from α_i^t would contradict the one-wayness of H. Hence, one can construct a probabilistic polynomial-time adversary C1 such that

$$|Pr[\text{A wins in G1}] - 1/2| \leq Adv^{\mathrm{OWF-H}}(\mathrm{C1},\lambda) + \mathrm{negl}(\lambda).$$

Combining the above inequalities, we obtain the stated bound. Therefore, the proposed construction satisfies backward secrecy. □

7.2. Security Under the ICR-3layAP Game

In this section, we prove that the proposed 3layAP protocol is secure with respect to the security game defined above. The goal is to show that the authentication-related ciphertexts generated in the three authentication layers are computationally indistinguishable from random bit strings.

Let the three authentication ciphertexts in the protocol be defined as follows:

- Layer 1 (login and session issuance): $C^{(1)}_{m,t} = f_{km}(St \parallel \sigma t)$;
- Layer 2 (policy-based role/permission verification): $C^{(2)}_{m,t} = f_{\sigma t}(St \oplus Rt)$;
- Layer 3 (group-membership authentication): $C^{(3)}_{i,t} = f_{GiK_t}(G_i_k_t)$.

Here, $C^{(1)}_{m,t}$, $C^{(2)}_{m,t}$, and $C^{(3)}_{i,t}$ denote the authentication ciphertexts generated in the first, second, and third layers, respectively. For the proof, we write the challenge ciphertext in the generic form

$$C^* = f_{K^*}(M^*),$$

where K^* is the challenge key material and M^* is the corresponding challenge plaintext. More specifically,

For $\ell = 1$, $K^* = km$ and $M^* = St \| \sigma t$;

For $\ell = 2$, $K^* = \sigma t$ and $M^* = St \oplus Rt$;
For $\ell = 3$, $K^* = GiK^t$ and $M^* = G_i_k_t$.
We assume the following.

Assumption 2-1. *The function f is a secure pseudo-random permutation (PRP) [16].*

Assumption 2-2. *The challenge key material used in the selected challenge instance is not directly exposed to the adversary, and the adversary cannot trivially obtain the exact challenge ciphertext through an allowed query to the game oracle.*

Under these assumptions, we prove the following theorem.

Theorem 4. *If f is a secure pseudo-random permutation and the key material is generated as specified in Section 6, then the protocol 3layAP is secure according to the security game ICR-3layAP. More precisely, for any probabilistic polynomial-time adversary A, there exists a probabilistic polynomial-time distinguisher B such that*

$$Adv_A^{ICR-3layAP}(\lambda) \leq Adv_{f,\mathrm{B}}^{PRP}(\lambda) + negl(\lambda).$$

Therefore, if f is PRP-secure, then the advantage of A in the game ICR-3layAP is negligible.

Proof. The proof is given by reduction to the PRP security of f. Suppose that there exists a probabilistic polynomial-time adversary A that wins the game ICR-3layAP with non-negligible advantage. We construct a distinguisher B that uses A to distinguish the PRP f from a truly random function.

The distinguisher B is given oracle access to $O(\cdot)$, where one of the following holds:

1. $O(x) = f_{K^+}(x)$ for a hidden random key K^+;
2. $O(x) = R(x)$ for a truly random function R.

The task of B is to determine which case applies. To do so, B simulates the game ICR-3layAP for A.

Setup.

B generates all public parameters honestly. It selects the hash function H, initializes the user set, role set, and group set, and creates user registration information. For each group Gi, it generates the initial secret values α_i^0 and β_i^0, and then derives the event-dependent values $\alpha_i^t = H^t(\alpha_i^0)$, $\beta_i^t = H^{q-t+1}(\beta_i^0)$.

Using these values, B computes the event group keys $GiK_t = f_{gi}(\alpha_i^t + \beta_i^t)$ for all non-challenge instances. The challenge instance is embedded through the oracle O, as described below.

Thus, the simulated setup is distributed identically to the real protocol setup, except that the challenge ciphertext will later be generated using the oracle O.

Queries.

The adversary A may issue the following queries: Registration Query, Login Query, Role-Verification Query, and Group-Authentication Query.

B answers these queries as follows.

- Registration Query:
  B registers the requested user honestly, including IDm, $H(PWm)$, role Rj, group number i, user secret key km, and the corresponding event group key GiK^t.
- Login Query:
  For any non-challenge instance, B computes $C_{m,t}^{(1)} = f_{km}(St \parallel \sigma t)$ honestly and returns it.
  For the selected challenge instance in Layer 1, B uses the oracle O.

- Role-Verification Query:
  For any non-challenge instance, B computes $C_{m,t}^{(2)} = f_{\sigma t}(St \oplus Rt)$ honestly. For the selected challenge instance in Layer 2, B uses the oracle O.
- Group-Authentication Query:
  For any non-challenge instance, B computes $C_{i,t}^{(3)} = f_{GiK^t}(G_i_k_t)$ honestly. For the selected challenge instance in Layer 3, B uses the oracle O.

Therefore, B simulates the entire query phase faithfully, except for the single challenge instance, which is answered through O.

Challenge.

The adversary A selects the authentication layer $\ell \in \{1, 2, 3\}$ and the corresponding challenge instance. According to the selected layer, B defines the challenge plaintext M^* as follows:

- If $\ell = 1$, then $M^* = St \| \sigma t$;
- If $\ell = 2$, then $M^* = St \oplus Rt$;
- If $\ell = 3$, then $M^* = G_i_k_t$.

B then queries the oracle O on M^* and returns $C^* = f_{K^*}(M^*)$ to A as the challenge ciphertext.

Now consider the following two cases:

- If O is a PRP oracle, then $C^* = f_{K^*}(M^*)$ and the challenge ciphertext is distributed exactly as in the real game ICR-3layAP.
- If O is a truly random function, then C^* is distributed as a random string of the same length, which matches the random challenge distribution in ICR-3layAP.

Hence, from the adversary's point of view, B provides either the real challenge distribution or the random challenge distribution, depending on whether O is a PRP or a random function.

Guess.

At the end of the game, A outputs a guess bit b', indicating whether the received challenge was a valid authentication ciphertext or a random bit string. The distinguisher B uses this output as its own guess for the nature of the oracle O:

- If A guesses that the challenge is real, then B guesses that O is a PRP oracle;
- Otherwise, B guesses that O is a random function.

If A has non-negligible advantage in distinguishing the real challenge from a random string in 3layAP, then B has the same non-negligible advantage in distinguishing the PRP oracle from a random oracle. Therefore,

$$Adv_A^{ICR-3layAP}(\lambda) \leq Adv_{f,\mathrm{B}}^{PRP}(\lambda) + negl(\lambda).$$

Since f is assumed to be a secure PRP, $Adv_{f,\beta}^{PRP}(\lambda)$ is negligible. It follows that $Adv_A^{ICR-3layAP}(\lambda)$ is also negligible. Therefore, 3layAP is secure according to the security game ICR-3layAP. □

Equivalently, the bound of Theorem 4 can be written in explicit form as $Adv_A^{ICR-3layAP}(\lambda) \leq Adv_{f,\mathrm{B}}^{PRP}(\lambda) + negl(\lambda)$. The reduction is tight: the distinguisher B performs exactly one oracle query for the challenge instance, so no multiplicative loss in the number of events q or in the number of queries arises. If the three layers are considered jointly, that is, if an adversary must distinguish the ciphertexts of all three layers at once, a standard hybrid argument over the layers replaces them by random strings one at a time, which yields the slightly looser bound $Adv_A^{ICR-3layAP}(\lambda) \leq 3 \cdot Adv_{f,\mathrm{B}}^{PRP}(\lambda) + negl(\lambda)$. Combining this with the

bounds of Theorems 2 and 3, the advantage against the three properties established in this section is bounded by $Adv_A^{total}(\lambda) \leq Adv_f^{PRP}(\lambda) + 2 \cdot Adv_B^{OWF-H}(\lambda) + negl(\lambda)$. The bound is therefore additive in the security of the underlying primitives and contains no factor that grows with the number of events or with the number of authentication requests, which matters in practice because q may be as large as 5×10^4 events per group and the protocol is intended for continuous online use (Section 8.3).

Theorem 4 shows that the protected authentication-related values in the proposed 3layAP protocol are computationally indistinguishable from random bit strings to any polynomial-time adversary. This result establishes confidentiality at the application layer. However, the ICR-3LAYAP security game captures confidentiality only, while integrity and authenticity of transmitted tokens are assumed to be provided by the underlying secure communication channel, such as TLS, used together with the proposed protocol. In addition, collusion resistance in group environments and resistance to inter-event key derivation are analyzed separately in Sections 7.1 and 7.3, respectively.

7.3. Security Against the Threat Model

In this section, we analyze how the proposed three-layer authentication protocol addresses the threat model defined in Section 5. The analysis is based on the two main results established in the previous subsections: (i) the event-based group key construction satisfies group key secrecy, forward secrecy, and backward secrecy, and (ii) the authentication ciphertexts generated at the three authentication stages are secure under the ICR-3layAP security game. Accordingly, the purpose of this section is not to introduce a new formal definition, but to explain how these proven security properties translate into resilience against the practical attack scenarios considered in the target group-oriented application environment.

7.3.1. Resistance to Impersonation and Unauthorized Access

The first threat concerns an adversary that attempts to impersonate a legitimate user or to access protected services beyond its assigned role and permission scope. In the proposed protocol, service access is not granted solely on the basis of account login. Instead, the user must successfully pass three consecutive authentication stages: login and session issuance, role and permission verification, and group-membership authentication. Therefore, an adversary must generate or forge valid authentication ciphertexts for all relevant stages in order to obtain unauthorized access.

More specifically, in the first layer, a valid login/session-related value of the form $C_{m,t}^{(1)} = f_{km}(St \parallel \sigma t)$ cannot be generated without knowledge of the long-term user secret km. In the second layer, without the session-related secret σt obtained from the first stage, the adversary cannot generate $C_{m,t}^{(2)} = f_{\sigma t}(St \oplus Rt)$. In the third layer, without the event-specific group key GiK_t, which is meaningful only after the preceding authentication stages have been successfully completed, the adversary cannot construct $C_{i,t}^{(3)} = f_{GiK_t}(G_i_k_t)$.

Hence, the adversary cannot bypass a lower layer and directly obtain access to group-protected services. This layered structure strengthens access control beyond conventional account-based authentication or policy-only RBAC enforcement. In AI-based enterprise document management environments, the protected assets include not only original documents but also chunks, embeddings, retrieval outputs, generated responses, and analysis results. Accordingly, it is necessary to verify not only account legitimacy but also the user's current role and group membership before service access is granted [22]. Therefore, the proposed protocol provides stronger resistance to impersonation and unauthorized access attempts.

7.3.2. Resistance to Replay and Token Misuse

The second threat concerns replay of previously observed authentication ciphertexts or misuse of tokens captured from prior sessions or events. In the proposed protocol, each authentication stage depends on the result of the previous stage. Thus, a value captured at an earlier point in time cannot be directly reused to reproduce the current authentication state.

In the first layer, the login result is bound to the session-related values St and σt, and therefore a ciphertext generated in a past session does not match the input required in the current session. The second layer is similarly tied to the role-verification result Rt and the session-dependent state, so a previously generated authorization value cannot be reused in a different session or for a different request. The third layer binds the group authentication process to the group index i, the event-specific group key GiK^t, and the authenticated state inherited from the earlier stages. Consequently, tokens from a previous event or from a different authentication stage cannot be replayed in the current group authentication procedure.

In addition, the authentication values of the three layers are not generated independently. Rather, each layer uses secret material derived from the previous layer. Therefore, even if an adversary captures a token from one stage, that token alone is insufficient to construct a valid token for another stage. This design prevents not only straightforward replay, but also cross-layer token misuse, i.e., the use of authentication material obtained at one layer to bypass another. As a result, the proposed protocol provides both session binding and stage binding against replay and token misuse.

7.3.3. Resistance to Collusion in Group Environments

The third threat concerns collusion attacks in group environments. In this setting, a malicious user may cooperate with a legitimate group member and obtain the current event-related group authentication material in an attempt to access services outside its own authorization scope. In conventional group-oriented authentication settings, exposure of valid group material may affect multiple accesses within the same group, and therefore collusion must be explicitly considered in dynamic group environments [18].

In the proposed protocol, however, the group authentication stage is not executed independently. It is performed only after the user has successfully completed both the first-layer login/session authentication and the second-layer role/permission verification. Hence, even if an adversary obtains the current event-specific group key GiK^t, this value alone is insufficient unless it is accompanied by a valid session state and a valid role-verification result associated with the same user context. Therefore, disclosure of current group-related material by a legitimate group member does not immediately enable full service access by an outside adversary. In this sense, the proposed design provides stronger protection than schemes in which possession of a group key alone is treated as the sole access condition.

Moreover, because the event-based group key is derived from the combination of a normal hash chain and a reversed hash chain, exposure of the current event-related material does not allow derivation of past or future event-specific group keys. Thus, even when colluding users share current group-related material, the impact of that disclosure is cryptographically confined to the current event and cannot be extended to prior or subsequent events. This temporal confinement is particularly important in dynamic membership environments.

It should be noted, however, that the collusion resistance considered here must be interpreted within a precise scope. The protocol guarantees that disclosure of current event-related group authentication material does not expand into derivation of past or future event keys. By contrast, if a legitimate user intentionally transfers its entire valid authentication state, including its active session state and role-verification state, to another party, such

behavior cannot be completely prevented by cryptographic mechanisms alone. In such cases, additional operational controls, such as audit logging, device binding, or abnormal-behavior detection, are required. Therefore, the collusion resistance of the proposed protocol should be understood as resistance against structural and temporal expansion caused by sharing group authentication material, rather than as a universal defense against all forms of credential sharing.

7.3.4. Resistance to Inter-Event Key Derivation

The fourth threat concerns an adversary that attempts to derive group keys of other events from group-related material observed at one event. This threat is particularly important in group environments because membership changes over time may alter the set of users authorized to access group resources. If the current event-related material allowed derivation of past or future event-specific group keys, then time-dependent group access control would effectively collapse.

In the proposed scheme, the event-specific group key at event t is defined as $GiK^t = f_{gi}(\alpha_i^t + \beta_i^t)$, where α_i^t is generated from a normal hash chain and β_i^t is generated from a reversed hash chain. Under this construction, knowledge of the current event-related material is insufficient to compute group keys of other events. In particular, deriving a future event key requires recovery of an unknown reversed-chain value, whereas deriving a past event key requires recovery of an unknown prior value from the normal hash chain. Each of these tasks amounts to solving a preimage problem for the hash function H, which is computationally infeasible under the one-wayness assumption [19,20].

Furthermore, if the adversary cannot recover the required hash-chain value and instead attempts to predict the corresponding PRP output directly, then the attack contradicts the pseudo-randomness of f [16]. Therefore, exposure of current event-related material does not imply computability of group keys for other events. In this sense, the proposed construction is structurally resistant to inter-event key derivation attacks. This property is a direct consequence of the forward secrecy and backward secrecy established in Section 7.1.

8. Efficiency Analysis

8.1. Evaluation Scope and Baselines

In this section, the efficiency of the proposed three-layer authentication protocol is evaluated from two complementary perspectives: theoretical analysis and implementation-based measurement. First, the structural characteristics of the proposed scheme are examined through time-complexity and storage-complexity analysis. Next, to complement the asymptotic analysis, a prototype implementation is used to measure stage-level authentication latency, total authentication latency per request, throughput, and the additional administrative overhead caused by membership changes.

It should be noted that the purpose of the proposed scheme is not to design an independent group-key management framework or to optimize large-scale rekeying itself. Rather, the objective of this work is to provide an integrated three-layer authentication structure for AI-based enterprise document management environments by combining login and session issuance, role/permission verification, and event-based group-membership authentication. Accordingly, the efficiency evaluation focuses primarily on the cost of the online authentication path that is repeatedly executed during normal service access.

For this purpose, three authentication configurations are considered as comparison baselines. Baseline 1 represents the basic structure that performs only account authentication and session issuance. Baseline 2 extends this structure by additionally performing role and permission verification after login and session processing. The final configuration is the full three-layer authentication structure proposed in this paper, which further incorporates event-

based group-membership authentication. Through this stepwise comparison, the additional cost introduced by each security layer can be examined explicitly, and the practical overhead of the proposed integrated authentication framework can be evaluated more clearly.

8.2. Time Complexity

In this section, we analyze the time complexity of the proposed three-layer authentication protocol by dividing it into the initialization stage and the online authentication stage. In the analysis, basic cryptographic operations such as hash, PRP, XOR, and concatenation are assumed to be constant-time operations. In particular, the proposed scheme adopts a structure in which all event-specific group keys are precomputed and stored in advance, rather than being generated on demand whenever a membership change occurs.

For consistency of the analysis, we use the following notation:

- *q:* the total number of events considered for a single group;
- *ni*: the number of members in group *Gi;*
- *M*: the total number of users;
- *G*: the total number of groups;
- *S*: the number of active sessions;
- *P:* the number of policy(rule) entries;
- *D*: the number of protected documents or objects.

8.2.1. Initialization/Preprocessing Cost

(1). Generation of Group Keys and Secret Values Required at Each Stage: $O(q)$.

- This step corresponds to the cost of pre-generating and storing all q event-specific group keys for one group in advance.
- If the total number of groups is G, the total system-wide cost is $O(Gq)$

(2). User Registration: $O(M)$.

- The total cost of assigning registration information, secret values, roles, and group information to all M users

8.2.2. Online Authentication Cost per Access Request

(1). Login and Session Issuance: $O(1)$.

(2). Policy-Based Role/Permission Verification: $O(1)$.

- However, if the policy table is implemented inefficiently using linear search, the worst-case complexity is $O(P)$.

(3). Group-Membership Authentication: $O(1)$.

- However, if the group member list is checked by linear scan, the worst-case complexity is $O(ni)$.

8.2.3. Final Time Complexity of the Overall Protocol

(1). Overall cost including initialization:

For a single-group setting, $O(q) + O(M) + O(1) + O(1) + O(1) = O(q + M)$.
That is, overall protocol complexity including preprocessing and registration is $O(q + M)$.
If the total number of groups G is also taken into account, then $O(Gq + M)$.

(2). Online cost for actual service access:

$$O(1) + O(1) + O(1) = O(1).$$

That is, overall online authentication complexity per access request is $O(1)$.

Since all event-specific group keys are precomputed and stored during the initialization stage, no additional event-key generation cost is incurred during online service access. Therefore, the online authentication phase consists only of constant-time verification and token-issuance procedures.

8.3. Storage Complexity

The proposed protocol adopts a structure in which all event-specific keys are precomputed and stored during the initialization stage. Accordingly, the storage complexity should be analyzed by taking into account not only user information and session states, but also the cost of maintaining the set of event-specific keys for each group.

On the user side, only the user identifier, password-related information, user secret value, role, group index, and the minimum authentication-related state need to be maintained. Therefore, the storage requirement per user is $O(1)$. On the server side, the system must store registration and authentication-verification information for all M users, as well as session states proportional to the number of active sessions S. In addition, since a total of q event-specific keys and their related states are stored in advance for each group, the storage cost for event-key material becomes $O(Gq)$ when the total number of groups is G. Therefore, the overall server-side storage complexity can be expressed as $O(M + Gq + S)$.

As a simple numerical example, assume G = 100 groups and q = 50,000 events per group over a five-year operational horizon, with each event-specific group key represented as a 128-bit value. In this case, the storage required for the precomputed event-key material is 100 × 50,000 × 16 bytes, i.e., approximately 80 MB. This example shows that the $O(Gq)$ storage term remains practically manageable at enterprise scale, while enabling $O(1)$ online authentication per request.

The proposed scheme requires additional storage space. However, this storage cost is fundamentally different from the cost of re-encrypting and re-storing the documents themselves. Rather, it corresponds to the cost of maintaining precomputed event-key material in advance for preserving group-specific authentication states. In other words, by using additional storage space, the proposed scheme eliminates the need to repeatedly perform document re-encryption, re-storage, and redistribution upon membership changes, thereby improving overall operational manageability in dynamic group environments.

8.4. Experimental Evaluation

To complement the asymptotic analysis in Sections 8.2 and 8.3, we conducted a prototype-based experimental evaluation of the proposed protocol. The purpose of this section is not to optimize each cryptographic primitive in isolation, but to verify that the proposed three-layer authentication structure is practically deployable under a standard implementation setting. The evaluation therefore focuses on three aspects: (i) preprocessing cost in the setup phase, (ii) end-to-end authentication latency across the three layers, and (iii) throughput and administrative overhead under membership change.

8.4.1. Prototype Implementation Environment and Experimental Settings

The prototype was implemented using standard cryptographic libraries on a single machine in a controlled environment. All timing results were obtained on the same hardware and software platform. To reduce the effect of initialization overhead and transient runtime noise, each experiment was first executed for 10 warm-up runs, and the actual measurement was then repeated 100 times. The reported execution time is the arithmetic mean of the 100 measured runs; when necessary, the standard deviation may also be reported.

Tables 1–3 present the experimental setup, cryptographic parameters and libraries, and experimental parameters.

Table 1. Experimental setup.

Hardware	CPU	11th Gen Intel(R) Core™ i5-1155G7 @ 2.50 GHz
	Memory	16 G
	Disk	1 TB
System type	64-bit operating system, ×64-based processor	
OS	Windows 11 Home	
Language	Python 3.15	

Table 2. Cryptographic parameters and libraries.

Cryptographic Parameters and Libraries	
Pseudo-random Permutation f	AES-128 operating as a 128-bit PRP on a single block.
Hash function h	SHA-256 truncated to128 bits
Crypto library	Cryptography
Python library	time, hashlib, secrets, concurrent.futures, etc.
Time measurement function	time.perf_counter()

Table 3. Experimental parameters.

Parameter	Description	Size/Value	Generation/Instantiation
km	Long-term secret key of user Um	128 bits	Randomly generated by a cryptographically secure random generator
gi	Master key of group Gi	128 bits	Randomly generated by a cryptographically secure random generator
St	Session identifier at time t	128 bits	Randomly generated for each session
σt	Session authentication token	128 bits	Randomly generated for each session
Rt	Role authentication token	128 bits	Randomly generated for each session
α_i^0	initial seed of the forward hash chain for group Gi	128 bits	Randomly generated by a cryptographically secure random generator
β_i^0	Initial seed of the reverse hash chain for group Gi	128 bits	Randomly generated by a cryptographically secure random generator
α_i^t	Forward hash-chain value for group Gi at event t	128 bits	Derived as $\alpha_i^t = H^t(\alpha_i^0)$

Table 3. *Cont.*

Parameter	Description	Size/Value	Generation/Instantiation
β_i^t	Reverse hash-chain value for group *Gi* at event *t*	128 bits	Derived from repeated hashing of β_i^0 and arranged in reverse event order
GiK_t	Event-specific group key of group *Gi* at event *t*	128 bits	Computed as $GiK_t = f_{gi}(\alpha_i^t + \beta_i^t)$
IDm	Identifier of user *Um*	Fixed encoded value	
Rj	Role identifier	Fixed encoded value	
q	Number of events	1000/10,000/100,000	
Concurrency level C	Number of simultaneous requests in throughput test	1/10/50	

In the prototype implementation, the secret values km, gi, St, σt, Rt and the initial seeds α_i^0 and β_i^0 were instantiated as 128-bit values. The random secret parameters were generated by a cryptographically secure random generator. The hash-chain values α_i^t and β_i^t were then derived by repeated application of the hash function $H(\cdot)$ from the corresponding initial seeds, where $\alpha_i^t = H^t(\alpha_i^0)$ and β_i^t was arranged in reverse event order from repeated hashing of β_i^0. The event-specific group key GiK^t was computed as $GiK^t = f_{gi}(\alpha_i^t + \beta_i^t)$. In the implementation, the keyed primitive f was instantiated as a 128-bit PRP, and the hash output was truncated to 128 bits so that the input and output lengths of the protocol values remained consistent across the preprocessing and online authentication procedures. The *Rj* was represented as fixed encoded value in the prototype, while the event parameter q was varied as 1000/10,000/100,000 in the preprocessing experiment.

For authentication-latency measurement, one trial corresponds to one complete execution of the relevant protocol stage or stage combination. For throughput measurement, the number of completed full authentication requests per second was recorded under single-request and under concurrent-request settings of 1, 10, and 50 clients. For membership-change evaluation, only protocol-intrinsic overhead was considered, namely server-side regeneration time, redistribution message count, and transmitted bytes. End-to-end network propagation delay was excluded because it mainly depends on deployment-specific network conditions rather than on the protocol design itself.

8.4.2. Authentication Latency Evaluation

We measured authentication latency at two levels. First, we measured the preprocessing cost of Section 6.1.1. In the preprocessing stage, this study focuses on the cost of generating event-specific group keys and the associated secret values, because this is the main offline operation whose cost grows directly with the number of events q. Other initialization tasks are comparatively lightweight and therefore omitted from separate measurement.

Second, we measured the latency of the actual authentication path experienced during service access. In this evaluation, Layer 1 latency denotes the elapsed time for login and session issuance, Layer 2 latency denotes the elapsed time for policy-based role/permission verification and recovery of Rt, and Layer 3 latency denotes the elapsed time for group-membership authentication, including the KDF-based derivation of the Layer 3 key. The total authentication latency is defined as the elapsed time for the complete three-layer authentication path.

Table 4 and Figure 2 show that the preprocessing cost increases almost linearly with the number of events q. The average time grows from 6.716 ms at q = 1000 to 71.794 ms at

q = 10,000 and 767.528 ms at q = 100,000, confirming a predictable increase in the dominant offline setup cost. Figure 2b further shows the per-event preprocessing cost, computed by dividing the total preprocessing time by q, and the resulting values remain nearly constant across different q. This indicates that the proposed preprocessing procedure preserves stable per-event efficiency even when a large number of event-specific group keys are generated in advance.

Table 4. Preprocessing cost for event-specific group-key generation.

q	Measured Preprocessing Task	Average Time (ms)	Per-Event Time (μs)
1000	Generation of event-specific group keys and required secret values	6.716	6.716 (6.716/1000 = 0.006716 ms = 6.716 μs)
10,000	Generation of event-specific group keys and required secret values	71.794	7.179 (71.794/10,000)
100,000	Generation of event-specific group keys and required secret values	767.528	7.675 (767.528/100,000)

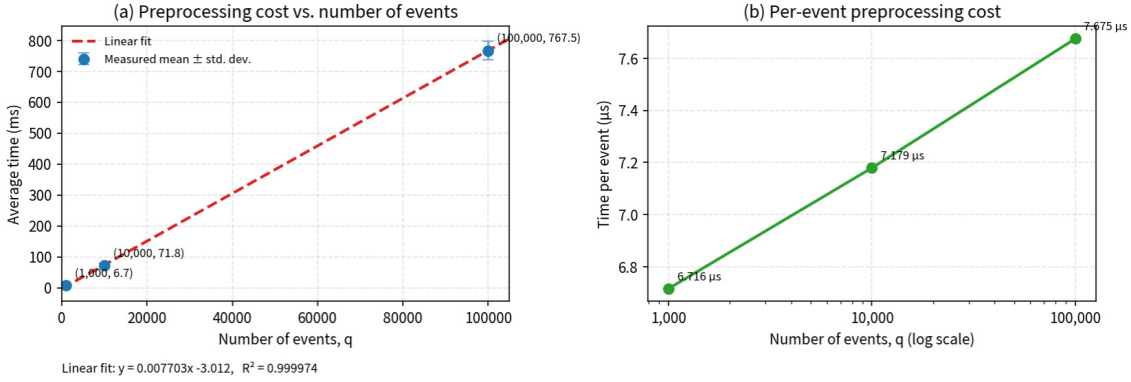


Figure 2. Preprocessing cost for event-specific group-key generation.

Table 5 shows that the proposed three-layer authentication path incurs only a very small online computational latency. The measured average latencies were 0.014 ms for Layer 1, 0.016 ms for Layer 2, and 0.020 ms for Layer 3, while the total authentication latency was 0.049 ms. Among the three stages, Layer 3 exhibited the largest latency because it includes both group-membership authentication and Layer 3 key derivation. However, the differences among the three stages remained small, indicating that the additional authentication logic does not impose excessive online overhead. Overall, these results support that the proposed protocol can provide multi-layer authentication while maintaining lightweight online performance. It should be noted, however, that these values were obtained in a controlled local prototype environment and reflect the computational latency of the protocol operations themselves. Therefore, they do not directly represent end-to-end user-perceived delay in a real deployment, where additional factors such as network transmission, system I/O, and service integration overhead may also contribute to the total response time.

Table 5. Stage-level authentication latency.

Measured Stage	Corresponding Procedure	Average Latency (ms)
Layer 1 latency	Login and session issuance	0.014
Layer 2 latency	Role/permission verification and recovery of Rt	0.016
Layer 3 latency	Group-membership authentication and Layer 3 key derivation	0.020
Total authentication latency	Complete three-layer authentication path	0.049

8.4.3. Throughput and Membership-Change Overhead

To further evaluate practicality, we measured the throughput of the complete online authentication path under single-request and light-concurrency settings. Throughput is defined here as the number of complete three-layer authentication requests processed per second.

Table 6 shows that the proposed protocol maintains high throughput over the complete online authentication path. The measured throughput was 19,516.041 requests/s at C = 1, 19,712.901 requests/s at C = 10, and 17,144.683 requests/s at C = 50. These results indicate that the proposed three-layer authentication procedure sustains stable online performance under low and moderate concurrency, with only a moderate decrease at the higher concurrency level. Since the measurements were obtained in a controlled local prototype environment, the absolute values may vary in actual deployment conditions.

Table 6. Throughput of the complete authentication path.

Concurrency Level C	Throughput (Requests/s)
1	19,516.041
10	19,712.901
50	17,144.683

Table 7 and Figure 3 show that the membership-change overhead increases linearly with the number of affected members. As the number of affected members increases from 10 to 100 and from 1000, the regeneration time rises from 0.049 ms to 0.518 ms and 5.219 ms, respectively, while the redistribution message count and the total transmitted bytes also increase in direct proportion to the group size. In particular, the regeneration cost per affected member remains nearly constant at about 4.9–5.2 µs, and the communication overhead is fixed at 64 bytes per member. These results confirm that the membership-change overhead follows the $O(ni)$ administrative cost analyzed in Section 8.2 and scales in a predictable manner. Since this cost occurs only when a join or leave event takes place, it should be interpreted as occasional management overhead rather than steady-state online authentication latency.

Table 7. Membership-change overhead.

Measurement Item	Meaning	Average Result (Affected Members)		
		10	**100**	**1000**
Regeneration time	Server-side time to regenerate affected authentication materials	0.049	0.518	5.219
Redistribution message count	Number of redistribution messages after a join/leave event	10	100	1000
Total transmitted bytes	Total communication volume for redistribution	640	6400	64,000

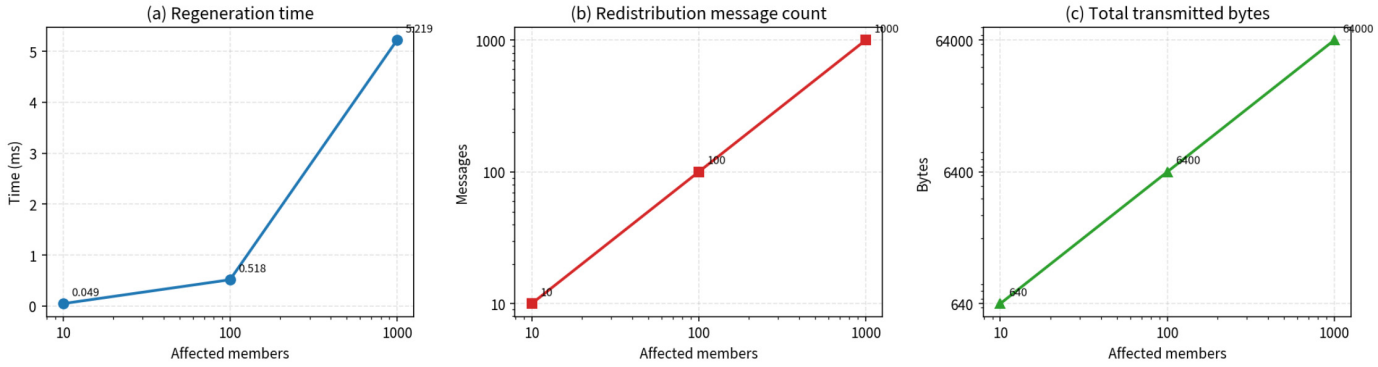


Figure 3. Membership-change overhead.

8.5. Comparative Discussion

Section 8.4 presented implementation-based results for preprocessing cost, authentication latency, throughput, and membership-change overhead. In contrast, this section clarifies the position of the proposed protocol through comparison with representative prior studies. Since the compared studies differ in purpose and application layer, it is not appropriate to judge them solely by absolute execution time. Therefore, the comparison focuses on four aspects: (1) how many security decision dimensions are integrated into online access control, (2) how membership changes are handled, (3) whether document re-encryption is required, and (4) which security functions are provided by recent AI-based document systems.

Table 8 compares the proposed protocol with representative traditional access-control and group-key-related studies. The RBAC model of Sandhu et al. [1] systematically established role-based permission management, but its online access decision is limited to one security dimension, namely role and permission verification. In contrast, this paper verifies session validity, role and permission legitimacy, and event-based group status before access is granted. Accordingly, the number of online access-control dimensions is expanded from one to three. Therefore, the proposed protocol provides more fine-grained document access control than the standalone RBAC model. However, RBAC remains advantageous in general information-system environments because of its simple structure and clear administrative model.

Table 8. Quantitative comparison with representative access-control and group-key studies.

Scheme	S	R	G	L	C_update	ReEnc
	Session Validity	**Role/ Permission**	**Event-Specific Group Legitimacy**	**L = S + R + G**	**Administrative Costs Associated with Membership Changes**	**Re-Encryption**
Sandhu et al. (Role-Based Access Control Models) [1]	0	1	0	1	N/A	No
Wong et al. (Secure Group Communications Using Key Graphs) [10]	0	0	1	1	$O(log\ n)$	Depends on application
Yang et al. (Reliable Group Rekeying) [26]	0	0	1	1	$O(log\ n)$-*class*	Depends on application
Proposed scheme	1	1	1	3	$O(ni)$	No

(Here, 0: No; 1: Yes).

The key-graph-based scheme of Wong et al. [10] and the reliable group rekeying study of Yang et al. [26] focus on group communication security and efficient key updates under membership changes. Since these studies are specialized for group-key management, they achieve lower rekeying cost than the proposed scheme. However, their focus is on group-key maintenance and distribution rather than on integrated access-control decisions. In this paper, optimal group-key rekeying is out of scope. Instead, the purpose is to combine the group key with the runtime authentication procedure of enterprise document services so that session, role, and group status can be verified together. Thus, although existing rekeying studies are superior in update optimization, the proposed protocol is more suitable for integrated authentication at the point of document access.

As shown in Table 8, the proposed protocol provides higher access-control precision than conventional RBAC or single-axis group-key-based approaches by simultaneously reflecting three security decision dimensions along the online access path. On the other hand, if only the administrative cost under membership change is considered, tree-based rekeying schemes such as those of Wong and Yang are more efficient. Therefore, the main strength of the proposed protocol lies not in rekeying optimization itself, but in its integrated authentication structure for online document access.

Table 9 shows that recent AI-based document system studies each focus on different security functions. The study by Byun et al. is strong in confidentiality filtering when external models are incorporated [27], the study by Lee et al. emphasizes the security of local or isolated execution environments [28], and the study by Aboukadri et al. focuses on the automation of policy extraction [29]. In contrast, the proposed scheme does not directly provide these functions, but it is distinguished by performing session authentication, role-based authorization, and dynamic group-state verification at the actual point of service access. Therefore, the proposed scheme should be understood not as a substitute for retrieval filtering, local deployment, or policy automation, but rather as a complementary protocol that strengthens the runtime access-control layer of AI-based enterprise document services.

Table 9. Quantitative comparison with recent AI-based secure document access studies.

Scheme	F	I	P	S	R	G	T
	Confidentiality /Security Filtering	**Isolated/Local Deployment Security**	**Policy Extraction /Automation**	**Runtime Session/User Authentication**	**Runtime Role-Aware Authorization**	**Runtime Dynamic Group/ Event-State Control**	**=F + I + P + S + R + G. Number of Security Features Provided Directly**
Byun et al. (Secure Multifaceted-RAG: Hybrid Knowledge Retrieval with Security Filtering) [27]	1	0	0	0	0	0	1
Lee et al. (Enhancing Security and Applicability of Local LLM-Based Document Retrieval Systems in Smart Grid Isolated Environments) [28]	0	1	0	0	0	0	1
Aboukadri et al. (Leveraging RAG and LLMs for Access Control Policy Extraction from User Stories in Agile Software Development) [29]	0	0	1	0	0	0	1
Proposed scheme	0	0	0	1	1	1	3

(Here, 0: No; 1: Yes).

9. Discussion

9.1. Token Lifecycle

The lifecycle of the protocol tokens is defined as follows. A session-related token is issued only after successful login authentication and remains valid only for the current active session. A role-related token is generated after policy-based role and permission verification and is bound to that authenticated session context. Likewise, a group-authentication token is newly issued for each session and is valid only within the corresponding session. Therefore, tokens generated in a previous session cannot be reused in a subsequent session.

Separately, changes in the group-event condition affect the underlying event-specific group key rather than the session-bounded token itself. When the relevant event state changes, the corresponding group key is updated according to the protocol design, and subsequent session-level group-authentication tokens are generated on the basis of the updated group key. Any token associated with an expired or terminated session is rejected and must be re-issued through the corresponding authentication procedure.

9.2. Salting, MFA, Challenge–Response, Implementation Validation

The primary focus of this study is the design of a three-layer authentication and authorization structure for AI-based enterprise document environments. Therefore, the present work does not attempt to exhaustively incorporate every deployable account-hardening mechanism within the protocol description. In practice, the proposed framework is compatible with additional measures such as salted credential storage, multi-factor authentication, and challenge–response techniques at the login layer. These mechanisms are complementary to the proposed design rather than contradictory to it.

9.3. Insider Threats, Server Compromise, Attack Points

The proposed protocol is designed to reduce the risk of unauthorized access by partially privileged insiders as well as external adversaries. In particular, access to a protected resource is not granted solely on the basis of login status or role possession, but additionally requires satisfaction of the current group-event condition. Thus, an insider who possesses only partial authorization information cannot directly bypass the remaining verification layers.

With respect to server-side exposure, the proposed protocol uses fresh one-time authentication values that are valid only within the current session during execution of the three-layer authentication procedure. Once the session changes, the previously stored values become invalid and cannot be reused, since new authentication tokens must be generated for all three layers. Moreover, a new session requires the user to enter the password again, while the server stores only its hash value. Under the one-wayness assumption of the hash function, disclosure of the stored password hash does not allow an adversary to recover the original password. Hence, exposure of prior authentication values and stored password hashes does not by itself enable successful authentication in a subsequent session.

Nevertheless, if a server-side attack occurs during an ongoing authenticated session, for example after Layer 1 and Layer 2 have been successfully completed but before the full protocol execution has terminated, the currently active session may be exposed to hijacking or misuse. This effect is limited to the live session which is attacked, because the protocol does not permit persistent reuse of authentication values across different sessions. Therefore, the proposed design mitigates the cross-session impact of token exposure, while not claiming protection against compromise of an already active session context.

9.4. Performance Bottlenecks in Large-Scale, High-Concurrency Deployment

The measurements in Section 8.4 were obtained on a single prototype node under light concurrency (up to 50 clients). Extrapolating them to an enterprise deployment identifies four practical bottlenecks that define the operating envelope of the proposed protocol, none of which arises from the cryptographic operations themselves.

The first is the centralization of the administrative authority. All three layers terminate at SA, so the authentication path can be no more scalable than the SA deployment. At the measured throughput of about 17,145 requests/s at C = 50, a single node is adequate for a mid-sized organization, but a large enterprise that concentrates several thousand accesses per second in the first minutes of the working day would saturate one instance. In practice the SA functions must be replicated behind a load balancer, and the session, role, and group-event state must be moved to a shared, replicated store. That store access adds one network round trip, typically on the order of 0.1–1 ms, which is one to two orders of magnitude larger than the 0.049 ms of protocol computation; the protocol cost then becomes a negligible fraction of the end-to-end per-request cost, and the state store, not the protocol, dominates.

The second is contention and logging on that shared state. Because every request is verified against the current session, role, and event state, concurrent requests belonging to the same session contend on a single record and can serialize if the store performs per-record locking; per-layer audit logging adds a write for every request, so at 10^4 requests/s the system must sustain 10^4 audit records/s. Batching, append-only log storage, and a retention policy are therefore required, and the audit path rather than the authentication path becomes the I/O bottleneck.

The third is the replication of the precomputed key material. As analyzed in Section 8.3, the requirement is $O(Gq)$, for example about 80 MB for G = 100 groups and q = 5×10^4 events at 128 bits per key. This is modest for one node, but a memory-resident key table must be present on every SA instance, so the footprint grows with the replication factor, and key rotation must be versioned so that instances do not verify against different key versions during the transition.

The fourth is the burst behavior of membership changes. The regeneration cost is $O(ni)$, measured at about 4.9–5.2 µs per affected member, which corresponds to roughly 50 ms and 640 KB of redistribution traffic for a group of 10,000 members. Under ordinary churn this overhead is negligible, but synchronized events such as a semester rollover or an organizational restructuring can trigger many large-group updates at once. Such updates should be queued and applied asynchronously, and the event state of an affected group should be advanced only after redistribution has completed, so that superseded group keys are not accepted during the transition.

Taken together, the protocol is computation-light but state-heavy: its scalability is governed by the replication, consistency, and I/O characteristics of the session, key, and audit stores rather than by the cost of the cryptographic operations. This is a favorable property for deployment, because such bottlenecks can be addressed with standard distributed-systems techniques such as sharding, read replicas, and asynchronous replication.

10. Conclusions

This paper proposed a three-layer authentication protocol for secure access control in AI-based enterprise document management environments. The main goal of the study is to strengthen enterprise access control by extending conventional login- and role-based protection with an additional group-level authentication layer suitable for collaborative settings such as project teams.

To achieve this goal, the proposed scheme integrates session authentication, role/permission verification, and event-based group-membership authentication into a unified structure. It uses event-specific group keys only as authentication material, not for document re-encryption, and the security analysis shows that the protocol strengthens access-control consistency under the stated threat model and security requirements.

Although an additional authentication layer is introduced, the protocol remains practically efficient at the application level because the first and second layers rely on lightweight operations and the group key is used only for authentication. The experimental results in Section 8 show that the added overhead is manageable in practical enterprise environments.

Although the analytical discussion, security proofs, and prototype-based measurements support the practicality of the proposed protocol, broader empirical validation remains an important direction for future work. Larger-scale implementation and attack-oriented evaluation under insider-threat and server-side stress scenarios will further strengthen deployment-level validation.

Funding: This study was supported by research fund from Honam University, 2025.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L.; Youman, C.E. Role-Based Access Control Models. *Computer* **1996**, *29*, 38–47. [CrossRef]
2. Mohamed, A.K.Y.S.; Auer, D.; Hofer, D.; Merz, M.; Breu, R. A Systematic Literature Review for Authorization and Access Control: Definitions, Strategies and Models. *Int. J. Web Inf. Syst.* **2022**, *18*, 156–190. [CrossRef]
3. Farhadighalati, N.; Estrada-Jimenez, L.A.; Nikghadam-Hojjati, S.; Barata, J. A Systematic Review of Access Control Models: Background, Existing Research, and Challenges. *IEEE Access* **2025**, *13*, 17777–17806. [CrossRef]
4. Chen, B.; Taipalus, T. Metadata-Aware RAG for Enforcing Access Control and Metadata-Based Filtering: Proof of Concept and Evaluation. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*; Association for Computing Machinery: New York, NY, USA, 2025. [CrossRef]
5. Jeong, J.; Lee, S.G. Permission-Aware RAG: Identity and Access Management (IAM)-Based Access Filtering in Multi-Resource Environments. *IEEE Access* **2025**, *13*, 192819–192835. [CrossRef]
6. Feng, D.; Cui, W.; Jiang, Y.; Yu, W.; Li, D. From Static Roles to Context-Aware Decisions: Integrating LLMs and RAG Into Access Control Frameworks for Power Systems. *IEEE Access* **2026**, *14*, 55864–55876. [CrossRef]
7. Ahmad, J.; Latif, S.; Khan, I.U.; Alshehri, M.S.; Khan, M.S.; Alasbali, N.; Jiang, W. An Interpretable Deep Learning Framework for Intrusion Detection in Industrial Internet of Things. *Internet Things* **2025**, *33*, 101681. [CrossRef]
8. Iqal, Z.M.; Selamat, A.; Krejcar, O. A Comprehensive Systematic Review of Access Control in IoT: Requirements, Technologies, and Evaluation Metrics. *IEEE Access* **2023**, *12*, 12636–12654. [CrossRef]
9. Yalli, J.S.; Hasan, M.H.; Jung, L.T.; Yerima, A.I.; Aliyu, D.A.; Maiwada, U.D.; Al-Selwi, S.M.; Shaikh, M.U.R. A Systematic Review for Evaluating IoT Security: A Focus on Authentication, Protocols and Enabling Technologies. *IEEE Internet Things J.* **2025**, *12*, 18908–18928. [CrossRef]
10. Wong, C.K.; Gouda, M.; Lam, S.S. Secure Group Communications Using Key Graphs. In *Proceedings of the ACM SIGCOMM Computer Communication Review*; Association for Computing Machinery: New York, NY, USA, 1998. [CrossRef]
11. Alsheavi, A.N.; Hawbani, A.; Othman, W.; Wang, X.; Qaid, G.; Zhao, L.; Al-Dubai, A.; Zhi, L.; Ismail, A.; Jhaveri, R.; et al. IoT Authentication Protocols: Challenges, and Comparative Analysis. *ACM Comput. Surv.* **2025**, *57*, 116. [CrossRef]
12. Shahidinejad, A.; Abawajy, J. An All-Inclusive Taxonomy and Critical Review of Blockchain-Assisted Authentication and Session Key Generation Protocols for IoT. *ACM Comput. Surv.* **2024**, *56*, 1–38. [CrossRef]
13. Ansari, S.A.; Ali, S. A Systematic Review of Lightweight Cryptographic Schemes for Security and Privacy in IoT. *Discov. Comput.* **2025**, *28*, 266. [CrossRef]
14. Ghaleb, B.; Ahmad, J.; Al-Dubai, A.; Khan, M.K.; Latif, S.; Khan, M.S. Lightweight Authentication Protocols for Secure IoT Communication Networks: A Comprehensive Survey, Taxonomy, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2026**, *28*, 6282–6317. [CrossRef]

15. Feng, P.; Li, B.; Han, B.; Ma, Y.; Hu, Y.; Zhao, R. A Scalable Group Authentication Protocol for IoT Based on PUF-Derived Bases in Inner Product Spaces. *Cybersecurity* **2026**, *9*, 41. [CrossRef]
16. Luby, M.; Rackoff, C. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM J. Comput.* **1988**, *17*, 373–386. [CrossRef]
17. Krawczyk, H.; Eronen, P. HMAC-Based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869. 2010. Available online: https://www.rfc-editor.org/rfc/rfc5869 (accessed on 1 November 2026).
18. Barker, E.; Chen, L.; Davis, R. *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*; NIST Special Publication 800-56C Rev. 2; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. Available online: https://csrc.nist.gov/pubs/sp/800/56/c/r2/final (accessed on 1 November 2026).
19. Rafaeli, S.; Hutchison, D. A Survey of Key Management for Secure Group Communication. *ACM Comput. Surv.* **2003**, *35*, 309–329. [CrossRef]
20. Fischlin, M. Fast Verification of Hash Chains. In *Topics in Cryptology—CT-RSA 2004; Lecture Notes in Computer Science 2964*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 339–352.
21. Kogan, D.; Manohar, N.; Boneh, D. T/Key: Second-Factor Authentication From Secure Hash Chains. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*; Association for Computing Machinery: New York, NY, USA, 2017.
22. Bellare, M.; Canetti, R.; Krawczyk, H. Pseudorandom Functions Revisited: The Cascade Construction and Its Concrete Security. In Proceedings of the 37th Symposium on Foundations of Computer Science (FOCS), Burlington, VT, USA, 14–16 October 1996.
23. Chen, B.; Tackman, J.; Setälä, M.; Poranen, T.; Mikkonen, T. Integrating Access Control with Retrieval-Augmented Generation: A Proof of Concept for Managing Sensitive Patient Profiles. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25)*; Association for Computing Machinery: New York, NY, USA, 2025. [CrossRef]
24. Republic of Korea. Enforcement Decree of the Personal Information Protection Act. Available online: https://elaw.klri.re.kr/eng_service/lawView.do?lang=ENG&hseq=60635 (accessed on 1 November 2026).
25. Barker, E. *Recommendation for Key Management: Part 1—General*; NIST Special Publication 800-57 Part 1 Rev. 5; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. [CrossRef]
26. Yang, Y.R.; Li, X.S.; Zhang, X.B.; Lam, S.S. *Reliable Group Rekeying: A Performance Analysis*; Technical Report; The University of Texas at Austin: Austin, TX, USA, 2001.
27. Byun, G.; Lee, S.; Choi, N.; Choi, J.D. Secure Multifaceted-RAG: Hybrid Knowledge Retrieval with Security Filtering. *Information* **2025**, *16*, 804. [CrossRef]
28. Lee, K.; Yang, S.; Jeong, J.; Lee, Y.; Shin, D. Enhancing Security and Applicability of Local LLM-Based Document Retrieval Systems in Smart Grid Isolated Environments. *Electronics* **2025**, *14*, 3407. [CrossRef]
29. Aboukadri, S.; Ouaddah, A.; Mezrioui, A.; El Asri, I. Leveraging RAG and LLMs for Access Control Policy Extraction from User Stories in Agile Software Development. *IEEE Access* **2025**, *13*, 116462–116472. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.