International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

TRACEABLE AND FORWARD-SECURE ATTRIBUTE-BASED SERVER AIDED VERIFICATION SIGNATURE SCHEME

Mridul Kumar Gupta
Department of Mathematics,
Chaudhary Charan Singh University,
Meerut- 250004, Uttar Pradesh India

Shivani Goel
Department of Mathematics,
Chaudhary Charan Singh University,
Meerut- 250004, Uttar Pradesh, India
Department of Mathematics,
C.L. Jain College, Firozabad 283203,
Uttar Pradesh, India

Abstract—**Attribute-based signature (ABS) emerges as a flexible and valuable cryptographic technology. Within an ABS framework, each signer receives a unique signing secret key based on their attributes and signs a message according to a specified signing policy aligning with their attributes. The verifier confirms that the signature originates from a signer whose attributes align with the signing policy. The verification process in most existing ABS schemes imposes a substantial computational burden. These involve numerous pairing operations and each pairing is expensive. Consequently, this poses a significant challenge for users with limited computational resources. In response to these challenges, we present a traceable and forward-secure attribute-based server aided verification signature scheme. It allows the verifier to authenticate the signature with the aid of an external cloud server. It involves assigning a significant portion of calculations to a powerful server in order to speed up the verification stage of the signature method. The proposed scheme alleviates the calculation burden on users as cloud servers do maximum computations and offers a more adaptable access policy.**

Keywords—**Attribute based signature; Traceable signature; Forward Security; Cloud Server; Server Aided verification**

I. INTRODUCTION

ABS stands out as a versatile and invaluable cryptographic technology. Under the ABS framework, each signer is provided with a distinct signing secret key based on their attributes, enabling them to sign a message in accordance with a predetermined signing policy that matches their attributes. The verifier ensures that the signature is from a signer whose attributes are corresponding to the signing policy. ABS has emerged as a versatile cryptographic technology with a wide range of important applications. These applications include but are not limited to private access control. It ensures selective access to resources based on predefined attributes; anonymous credential issuance which enables users to authenticate themselves without revealing their identities; trust negotiations, where parties exchange attributes to establish trust before engaging in transactions; distributed access control. It decentralizes access management across multiple entities or devices; and attribute-based messaging, which facilitates secure communication based on specific attributes of users or entities involved. The flexibility and adaptability of ABS make it a valuable tool in various domains where secure authentication, access control, and communication are crucial. Majiet. al. [16] initially established the formal definition of ABS, laying the foundational groundwork for subsequent research in this field. Furthermore, they developed a concrete ABS technique & rigorously demonstrated the security guarantees of their proposed technique through rigorous analysis and proofs. A traceable attribute-based signature (TABS) is another cryptographic technique that allows for the tracing of signatures back to their originators in a secure and privacy-preserving manner. This type of signature scheme enables authorities or designated entities to trace the source of a signature without compromising the privacy of the signers or revealing their identities to unauthorized parties. These schemes find applications in various scenarios where accountability and

traceability of signatures are essential, such as in digital transactions, authentication protocols, and electronic voting systems. Escala et al. [7] presented a novel traceable attribute-based signature (TABS) technique. This technique incorporates innovative techniques, such as utilizing structure-preserving signatures, to enable the traceability of signers while preserving the integrity of the signature structure. In order to address the potential risk of key exposure within ABS schemes, Yuenet al. [23] presented the concept of forward secrecy of signatures (FSS) during the establishment phase of ABS systems. This notion of FSS aims to enhance the security of ABS schemes by ensuring that even if a signer's secret key is compromised in the future, signatures generated with that key prior to the compromise remain confidential and cannot be forged by adversaries. Yuen et al.'s [23] proposal contributes to mitigate the vulnerabilities associated with key exposure in ABS systems, thereby bolstering the overall security and integrity of digital signature protocols. However, one of the primary limitations of ABS is the substantial computational burden associated with the verification process. In certain existing ABS schemes, the verification step demands a significant number of pairings, which typically increases in direct proportion to the size of the predicate formula. It's worth noting that pairings are computationally expensive operations, further exacerbating the computational cost of the verification procedure. So, in this paper we present the traceable and forward-secure attribute-based server-aided verification signature scheme, which encompasses constant-size support for flexible threshold predicates. This innovative scheme combines traceability and forward security features with the assistance of an external server to authenticate signatures. Under the scheme, the verifier is empowered to verify signatures with the aid of an external server, enhancing the authentication process. This collaborative verification approach not only ensures the integrity of the signatures but also mitigates the computational burden on users.

II. RELATED WORK

Attribute based signature: Maji et al. [16] initially established the official definition of ABS and presented a concrete ABS scheme, accompanied by a thorough proof of its security. However, Li et al. [12] later identified weaknesses in the security of the technique proposed by Maji et al. [16], highlighting that the security proof was based on the generic group model. In response, Guo et al. [18] introduced a new ABS technique. Nonetheless, Maji et al. [17] pointed out that this scheme lacked consideration for signer privacy in its security definition. To address this concern, Maji et al. [17] devised an ABS technique that ensures perfect privacy. They validated the security of their approach by leveraging non interactive proof systems and incorporating a credential bundle scheme to conceal users' attributes. Zhang et al. [46] discovered vulnerabilities in Maji et al.'s [17] third instantiation, revealing that it was susceptible to forgery attacks. Consequently, Zhang et al. [24] unveiled their forgery attack against Maji et al.'s [17] scheme and elucidated the underlying reasons behind the vulnerability. Escala et al. [7] introduced a Traceable Attribute-Based Signature (TABS) scheme, emphasizing the traceability of signers through the use of structure-preserving signatures and substantiating the security of the technique. Additionally, El Kaa farani et al. [6] brought attention to the inefficiency of the scheme [7] due to its reliance on the composite-order groups setting. In response, ding et al. [5] proposed an efficient TABS technique that strikes a balance between traceability and privacy. Their construction is supported by the existential unforgeability proof, utilizing the q-augmented Diffie-Hellman exponent assumption. This advancement addresses the shortcomings of the previous scheme [7] by enhancing efficiency without compromising security. Additionally, Lu et al. [15] contributed to the field by presenting a traceable attribute-based signcryption scheme using threshold predicates. Their scheme is utilized in real-world applications like in healthcare social networks and offers fixed size signcryption and demonstrates practicality. To address the risk of key exposure in ABS schemes, Yuen et al. [23] introduced the concept of Forward Secrecy of Signatures (FSS) during the setup phase of ABS. Wei et al. [19] introduced a Forward-Secure Attribute-Based Signature (FS-ABS) technique that supports threshold predicates. They began by formalizing security definitions for their proposed technique, focusing on forward security and attribute signer privacy. The procedure involved updating the signing key at various time intervals using a binary tree structure, a methodology often suggested for constructing hierarchical Identity-Based Encryption (IBE) techniques [2, 1]. For adjustable threshold predicates, [11] offered a constant-size TFS-ABS scheme that enables forward security and traceability.

Server-Aided Verification Signature: The primary issue with numerous current ABS methods [14, 13, 22, 21, 9, 6, 5, 15, 18] is the high computation overhead in the verification algorithm. Devices with limited resources shouldn't use it. Sever-aid verification employs a strong server to assist devices with restricted resources in carrying out cryptographic operations in order to address this problem. It's the best method for cutting down on computation overhead. In actuality, though, the user is more likely to deal with a semi-trusted service that tries to compromise their privacy or provides incorrect results. Jakobsson et al. [10] originally introduced the criteria of server-aid signature in order to stop the semi-trusted server. More broadly, a server aided verification approach was offered by Girault and Lefranc [8]. A formal explanation of the security model to achieve collusion attacks was given by Wu et al. [20]. A particular server-aided verification signature approach that is resistant to collusion attacks was presented by them.

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

Nevertheless, user anonymity cannot be safeguarded by server-aid verification signatures. Li et al. [3] used Wu et al.'s approach to provide an outsourced verification ABS scheme that ensured anonymity. In this paper, we present the Traceable and Forward-Secure Attribute-Based Server-Aided Verification Signature scheme. [4] also provided an ABSAVS scheme for devices with limited resources to safeguard users' data integrity. The suggested scheme incorporates both traceability and forward security features, leveraging the assistance of an external server to enhance the security and reliability of signature authentication processes. By integrating these advanced security measures, the scheme ensures the integrity and confidentiality of signatures, thus bolstering the overall security of the authentication process.

Our Contribution

In this paper, we present the traceable and forward-secure attribute-based server aided verification signature scheme. The suggested scheme integrates traceability and forward security features with the assistance of an external server to enhance the verification process. With this scheme, the verifier gains the ability to verify signatures with the support of the external server, streamlining the authentication process and ensuring the integrity of the signatures within less time. By adopting a collaborative verification approach, the scheme alleviates the computational burden on users as compared to the original verification technique. This innovative solution significantly improves the efficiency and effectiveness of signature verification in resource constrained devices and in various applications and scenarios.

Paper Organization

The subsequent sections of our paper are organized as follows: In Section 3, we present a brief introduction to relevant concepts and a table of notations with their meanings. Section 4 outlines the framework, security model, and system architecture of our scheme. In Section 5, we detail the step-by-step construction of our signature technique. Section 6 covers the correctness and security proofs of our scheme, along with a comparative evaluation against existing signature schemes, highlighting distinctive features and advantages. Finally, in Section 7, we provide concluding thoughts and summarize the key findings and contributions of the article, followed by the references.

III. PRELIMINARIES

3.1 Lagrange Interpolation

There is a unique polynomial D with real coefficients fulfilling $D(x_i) = z_i$ for $i \in \{1,2,\dots,m\}$
s.t. deg(D) <m for given m different real values $x_1, x_2, \dots, x_m$and m real values $z_1, z_2, \dots, z_m$ (not basically different). This polynomial is computed by

$$D(x) = \Sigma_{j=1}^{m} D_j(x) \quad (1)$$

Where,

$$D_j(x) = z_j \Pi_{(k=1,k\neq j)}^{m} \frac{x-x_k}{x_j-x_k} \quad (2)$$

Bilinear Map

Suppose H_1,H_2 be two groups under addition and H_T be the group under multiplication. Each group is having prime order 'p'. A pairing 'e'

$$e: H_1 \times H_2 \rightarrow H_T$$

is bilinear map if it satisfies the following two properties:

1. Bilinearity:
For all $P_1 \in H_1, P_1 \in H_1$
and $c, d \in Z_p$
We have

$$e(cP_1, dP_2) = e(P_1, P_2)^{cd} \quad (3)$$

Also

$$e(P_1, P_2P_3) = e(P_1, P_2)e(P_1, P_3) \quad (4)$$

2. Non-degeneracy:
For $P_1 \neq 0_{H_1}$ *and* $P_2 \neq 0_{H_2}$, $e(P_1, P_2) \neq 1_{H_T}$

This bilinear property is equivalent to many equalities as:

$$e(cZ_1, dZ_2) = e(dZ_1, cZ_2) = e(cdZ_1, Z_2) = e(cZ_1, Z_2)d = e(dZ_1, Z_2)c = \cdots \quad (5)$$

is known as bilinear pairing.

3.2 Binary Tree

Canetti et al. [2] described the binary tree structure. In this structure, the whole time is divided into $T = 2^l$ discrete time intervals, or $t_0, t_1, \dots, t_{T-1}$. The leaf node of the entire l-deep binary tree is arranged from left to right chronologically and associated with each time period. The tree's root node is represented by the null string ϵ. An i-bit string $b_v \in \{0,1\}^i$represents a path with depth $i(1 \leq i \leq l)$ in this tree that runs from the root to any node v. The numbers 0 and 1 denote that the path travels through the left and right children of the antecedent node, respectively. In contrast, We use depth i to indicate the node for this binary tree by v_b for each string $b \in \{0,1\}^i$. We indicate that the length of the string b_v is $|b_v|$, and that the i-th bit of b_v is $b_v[i]$.

To illustrate, the top leaf node v_{t_0} matches the first time period t_0, as $b_{v_{t_0}} = 0^l$. Its right sibling v_{t_1} corresponds to the second time period t_1, as $b_{v_{t_1}} = 0^{l_1}1$. This is depicted in Fig. 1. The node set on the path (involving v and root node) from the root to the node v is what $Path_v$ is represented by. $R(v)$ is represented by $v's$ right child. For each time period t_i associated with a leaf node v_{t_i}, we define the set $V_{t_i} = \left\{v \in Path_{v_{t_i}} R(v) \notin Path_{v_{t_i}}\right\} \cup \{v_{t_i}\}$. As an

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

illustration, consider the binary tree with depth 3 in Fig.1, there exists $Path_{v_{t_0}} = \{\epsilon, v_0, v_{00}, v_{t_0}\}$, and $V_{t_0} = \{v_1, v_{01}, v_{t_1}, v_{t_0}\}$.

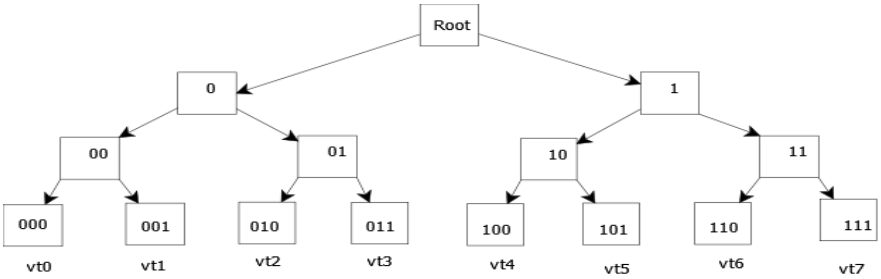

Figure 1: Binary Tree

3.3 Complexity Assumption

Our construction's security depends on the q-Diffie-Hellman exponent (q-DHE) hypothesis.

Definition: (q-DHE). We posit that the (t, ϵ) q-Diffie-Hellman exponent (q-DHE) hypothesis holds within a group G when there's no probabilistic polynomial-time adversary capable of solely computing $g^{a^{q+1}}$ based on the elements $\left(g, g^a, g^{a^2}, \ldots, g^{a^q}, g^{a^{(q+2)}}, \ldots, g^{a^{(2q)}}\right)$ Within a time, frame of t, with a probability exceeding ϵ. Here, $a \in Z_p$ and $g \in G$ are independently and uniformly selected. Alternatively, we define the (t, ϵ)-computational infeasibility of the q-DHE problem in G as the condition where any probabilistic polynomial-time (PPT) adversary A, operating within a time, frame of t, possesses an advantage $Pr\left[g, g^a, g^{a^2}, \ldots, g^{a^q}, g^{a^{(q+2)}}, \ldots, g^{a^{(2q)}}\right]$ that is less than or equal to ϵ.

3.4 Notations

In order to provide clarification on the notations used in our research, we have created Table 1.

Table 1: Notations

Notations	Meaning
$F_{k,\chi^{*(\cdot)}}$	The encryption access structure
W	The universal set of attributes
ξ	Cardinality of W
κ	dummy attribute set
I	Identity universe
I_d	Signer's identity.
t	A threshold value
χ	A set of attributes
R_1, R_2	Two groups of prime order p
e	The bilinear pairing
$q(\cdot)$	A polynomial of degree $t-1$
F	The hash function.
$\Delta_{i,V}(0)$	The Lagrange coefficient

IV. FRAMEWORK AND SECURITY MODEL

4.1 Framework

Our scheme consists of the following algorithms: Setup algorithm, KeyGen algorithm, Update algorithm, Sign algorithm, Transform algorithm, Server-aided verify algorithm, Lightweight-verify algorithm, Trace algorithm. These algorithms are defined as follows:

Setup: Attribute Authority (At_{Aut})performs the Setup algorithm's execution. The input parameters that it receives

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

are the security parameter λ, system threshold d, attribute universe U, total no. of time periods T, and dummy attribute set κ. The algorithm then outputs the master secret key MSK along with public parameter params.

KeyGen: The (At_{Aut}) algorithm runs the KeyGen algorithm. The signer's identity Id, where Id represents the universe of identities, the public parameter params, the master secret key MSK, and the signer's attribute set χ a subset of the attribute universe U plus the dummy attribute set κ are all entered. The current time period is also set by the method to $t_0 = 0$.

The tracing key tk and the signature secret key sk_{t_0} are the results of this procedure.

Update: The person in possession of the signing secret key performs the Update algorithm. The signature secret key sk_{t_i} for the current time period t_i and the following time period t_j where $t_i < t_j \leq t_{T-1}$, are entered into this procedure as inputs. The next output of the method is the signing secret key sk_{t_j} for the subsequent time period t_j.

Sign: The signer possessing an attribute set χ, runs the Sign algorithm. The public parameter, the signing predicate $F_{k,\chi^*}(\cdot)$, the current time period t_i, the signer's signing secret key sk_{t_i} corresponding to the attribute set χ, and a message M are all inputs where χ^* is subset of the attribute universe W, and $1 \leq k \leq |\chi^*| \leq d$.At time period t_i, the method generates a signature ν corresponding to the signing predicate $F_{k,\chi^*(\cdot)}$ if the attribute set χ meets the signing predicate $F_{k,\chi^*(\cdot)}$, which means that either $|\chi \cap \chi^*| \geq k \; or \; F_{k,\chi^*}(\chi) = 1$.

Transform: This transform algorithm is executed by the verifier. Upon receiving the signature, the verifier generates a transformed signature v'. This transformation involves calculating the transformed signature using a randomly chosen number a, which the verifier keeps secret.

Server-aided verify: This algorithm is executed by the server. Upon receiving the transformed signature from the verifier, the server proceeds to compute a token, denoted as Λ.

Lightweight-verify: This algorithm is executed by the verifier. In this algorithm, the inputs include the token Λ, the public key PK, and the predicate. The output is either true or false, depending on whether the signature is determined to be valid or invalid.

Trace:At_{Aut} uses the Trace method to determine the true identity of the signer from the signature v. The public parameter params, the message M, the signature v connected to the signing predicate $F_{k,\chi^*}(\cdot)$, and the tracing key tk are the inputs of this method. The true identity of the signer is then output.

4.2 Security Model

4.2.1 Traceability

Let the signature scheme that is traceable and forward-secure meets both privacy and forward security criteria. For the suggested scheme to be traceable, it must be possible for the At_{Aut} to determine the identity I_d of the signer for $(MSK, params) \leftarrow Setup(\lambda, t, S, W, \kappa)$, message M, attribute set χ, secret key $(sk_{t_i}, tk) \leftarrow KeyGen(MSK, params, t_i, I_d, \chi)$, and signing predicates $F_{\kappa,\chi}(\cdot)$ s.t. $F_{\kappa,\chi}(\cdot) = 1$, a time period t_i & signatures $v \leftarrow Sign(M, params, F_{\kappa,\chi}(\cdot), sk_{t_i})$.

4.2.2 Attribute signer privacy

When, for any given message M, signature ν on predicate F, and sets of attributes χ_1 and χ_2 such that $F(\chi_1) = F(\chi_2) = 1$, an adversary A cannot determine with greater accuracy than random guessing which attribute set, χ_1 and χ_2, was used in generating the signature ν, then the signature scheme satisfies attribute signer privacy.

4.3 System architecture

The proposed scheme's system model is shown in Fig. 2. It involves four entities: the attribute authority (At_{Aut}), the signer, the verifier, and the cloud server. In this model, (At_{Aut}) is responsible for managing the signer's attributes and it is trusted third party. Initially, (At_{Aut}) generates the master secret key (MSK) and public parameter (params), where the public parameter are publicly available. Additionally, Using MSK, the signer's attribute set (κ), and identification (I_d), (At_{Aut}) generates the signature secret key (sk_{t_0}) and tracing key (tk). Afterwards, (At_{Aut}) keeps tk private and gives (sk_{t_0}) to the signer. The signer updates the key (sk_{t_i}) for the current time period (t_i) to produce the signing secret key (sk_{t_j}) for the next time period (t_j), where $t_i < t_j \leq t_{T-1}$ and T signifies the total number of time periods. Finally, usingsk_{t_i} and a signing predicate (γ), the signer generates a signature (v_i) for the message (M) of the current time period (t_i). To confirm the accuracy of the signature v_i, the verifier seeks assistance from the cloud server to authenticate the signature created by the signer. The verifier forwards the transformed signature to the server, which then returns a token to the verifier. Utilizing this token, the verifier can easily verify the authenticity of the original signature. Additionally, the attribute authority (At_{Aut}) employs the tracing key t_k to monitor the authentic identity of the signer I_d in case of misuse of their signing behavior.

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

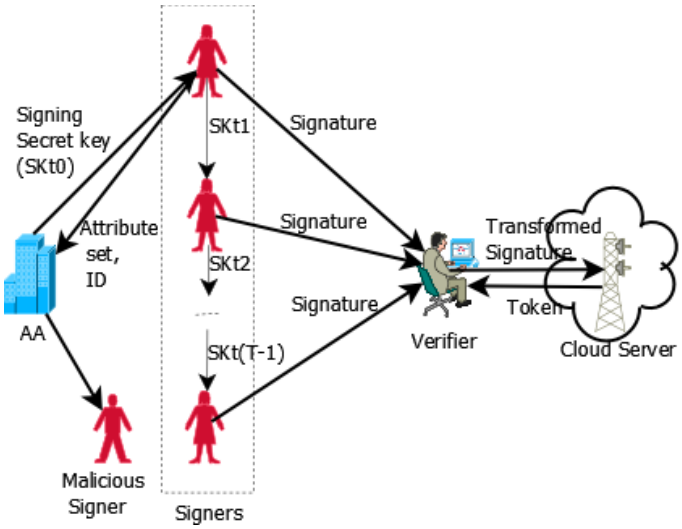

Figure 2: System architecture

V. OUR CONSTRUCTION

Setup: The algorithm establishes t as the system threshold and $S = 2^k$
as the total number of time periods, where k represents the binary tree's depth. This step accepts input λ as a security parameter & it establishes $I \subseteq Z_p$
be the identity universe $\&W \subseteq Z_p(|W| = \xi)$ as the attribute universe. Suppose $\kappa = \{\xi + 1, \xi + 2, \dots, \xi + t - 1\}$ represent a fake attribute set with $t - 1$ attributes, and let $W = \{1, 2, 3, \dots, \xi\}$. Additionally, one hash function $F: \{0, 1\}^* \rightarrow Z_p^*$ is selected by the method. Let R_1 *and* R_2 be two prime-order multiplicative cyclic groups. A bilinear pairing is $e: R_1 \times R_1 \rightarrow R_2$. A generator $g_1 \in R_1$ and a node$x \in Z_p^*$ are randomly selected using the algorithm, which then computes

$$Z = e(g_1, g_1)^x \quad (6)$$

The algorithm randomly selects $h', h_1, \dots, h_n, d_0, d_1, \dots, d_{\xi+t-1}, f_0, f_1, f_2, \dots, f_k$ from R_1, and sets the vectors

$$H = (h', h_1, \dots, h_n),$$
$$D = (d_0, d_1, \dots, d_{\xi+t-1}),$$
$$F = (f_0, f_1, f_2, \dots, f_k)$$

Eventually, the algorithm returns
$params = (W, \kappa, R_1, R_2, g_1, e, Z, H, D, F)$,
$MSK = x$
as public parameters and master secret key respectively.
KeyGen: For the attributes of signer at the first time period t_0, the procedure generates the signing secret key SK_{t_0}. The algorithm accepts as inputs MSK, params, t_0, $I_d \in I$, and $\chi \subseteq W$, where I_d denotes the identity of the signer and χ denotes the signer's attribute set.
(a) At_{Aut}determines other locations by selecting at random a (t−1) degree polynomial q(y) with q(0) = x & computes another points $q(i)$ where $i \in (\chi \cup \kappa)$.
(b) In order to determine the signer's identity Id & attribute set χ, At_{Aut} randomly selects μ_0, $h_0 \in Z_p$.
It then sets

$$y_0 = (h')\mu_0 . F(I_d),\ y_1 = g_1^{h_0},\ y_2 = e\left((h')\mu_0 F(I_d),\ g_1\right)$$

and

$$tk = y_3 = g_1^{\mu_0}$$

In this way, At_{Aut} maintains a list called (I_d, tk), &tk is used to trace the true identity of the signer.
(c) Each time an attribute $i \in (\chi \cup \kappa)$, At_{Aut} selects $\gamma_i \in Z_p$ at random. Furthermore, the method randomly selects $\gamma_{i,s} \in Z_p$ for every node $s \in T_{t_0}$. and finds

$$SK_{i,s} = \left(u_i, v_i, \{s \in S_{t_0}\}\right) \quad (7)$$

Where,

$$u_i = g_1^{\gamma_i},\ v_i = \left(d_1^{\gamma_i}, \dots, d_{i-1}^{\gamma_i}, d_{i+1}^{\gamma_i}, \dots, d_{\xi+t+1}^{r_i}\right)$$

$$w_{i,s} = g_1^{q(i)}(d_0 d_i)^{\gamma_i}\left\{f_0\left(\prod_{j=1}^{|b_s|} f_j^{b_s[j]}\right)\right\}^{\gamma_{i,s}},\ g_1^{\gamma_{i,s}},\ f_{|b_s|+1}^{\gamma_{i,s}},$$
$$f_{|b_s|+2}^{\gamma_{i,s}}, \dots,\ f_k^{\gamma_{i,s}}$$
$$= (\beta_{i,0}, \beta_{i,1}, \beta_{i,|b_s|+1}, \dots, \beta_{i,k}).$$

(d) The process eventually yields the original signing secret key
$SK_{t_0} = \{y_0, y_1, y_2, SK_{i,v}\}$
(8)

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

for the signer, where $i \in (\chi \cup \kappa)$ and $s \in T_{t_0}$

Update: For the subsequent time period t_j, the process yields the signing secret key SK_{t_j}. This algorithm runs in the following way:
(a) Parse the signing secret key SK_{t_i} as $\{i \in (\chi \cup \kappa)\}$, where

$$SK_{i,s} = \left(s \in T_{t_i}\right),$$
$$w_{i,s} = \left(\beta_{i,0}, \beta_{i,1}, \beta_{i,|b_s|+1}, \dots, \beta_{i,k}\right) \quad (9)$$

(b) Since $t_j \geq t_i$, we know that there is a node $s \in T_{t_i}$ for every node $s' \in T_{t_j}$ s.t. $b_{s'} = b_s \parallel b^*$ for some string b^*.
(c) The procedure chooses at random $\gamma_i' \in Z_p$ for each attribute $i \in (\chi \cup \kappa)$. Additionally, for each node $s' \in T_{t_j}$, the process chooses randomly $\gamma_{i,s'} \in Z_p$. Next, the process determines the component of the signing secret key.

$$SK_{i,s'} = \left(u_i', v_i', \left\{s' \in T_{t_j}\right\}\right) \quad (10)$$

where,

$$u_i' = u_i g_1^{\gamma_1'},$$

$$v_i' = d_1^{\gamma_i} d_1^{\gamma_i'}, \dots, d_{i-1}^{\gamma_i} d_{i-1}^{\gamma_i'}, d_{i+1}^{\gamma_i} d_{i+1}^{\gamma_i'}, \dots, d_{\xi+t+1}^{\gamma_i} d_{\xi+t+1}^{\gamma_i'}$$

$$w_{i,s'} = \beta_{i,0}(d_0 d_i)^{\gamma_i'} \left\{ f_0 \left(\prod_{j=1}^{|b_s'|} f_j^{b_{s'}[j]} \right) \right\}^{\gamma_{i,s'}}, \beta_{i,1} g_1^{\gamma_{i,s'}},$$
$$f_{|b_{s'}|+1}^{\gamma_{i,s'}}, \beta_{i,|b_s|+1} f_{|b_{s'}|+1}^{\gamma_{i,s}'}, \dots, \beta_{i,\kappa} f_\kappa^{\gamma_{i,s}}$$
$$= \left(\beta_{i,0}', \beta_{i,1}', \beta_{i,|b_{s'}|+1}', \dots, \beta_{i,k}'\right)$$

(d) This new signing secret key $SK_{t_j} = \{y_0, y_1, y_2, SK_{i,s'}\}$ is eventually returned by this process, where $i \in (\chi \cup \kappa)$ & $s \in T_{t_j}$. Furthermore, the signing secret key SK_{t_i} is deleted.

Sign: The attribute set χ must meet the predicate $\gamma_{\kappa,\chi^*}(\cdot)$ in order to endorse a message along with the following format: $M = (m_1 m_2 \dots m_n) \in \{0,1\}^n$ with predicate $F_{\kappa,\chi^*}(\cdot)$& the signing secret key sk_{t_i}. Alternatively, a k-elements subset $\chi' \subseteq (\chi \cap \chi^*)$exists. This algorithm continues as follows:

(a)The signer chooses a dummy attribute subset $\kappa' \subseteq \kappa$ associated having $(t-r)$ elements, then denotes $V = \chi' \cup \kappa'$ $(|S| = t)$.
The algorithm sets $\kappa = \{\xi+1, \xi+2, \dots, \xi+t-r\}$.

(b)Parse the signing secret key SK_{t_i}as $\{i \in (\chi \cup \kappa)\}$, where

$$SK_{i,s} = \left(s \in T_{t_i}\right),$$

$$w_{i,s} = \left(\beta_{i,0}, \beta_{i,1}, \beta_{i,|b_s|+1}, \dots, \beta_{i,k}\right). \quad (11)$$

Then, parse $SK_{i,s_{t_i}} = \left(\beta_{i,0}, \beta_{i,1}\right)$.
(c) For each attribute $i \in V$, the algorithm uses $SK_{i,s}$ to compute as follows.

$$\beta_{i,0}^* = \beta_{i,0} \left(\prod_{j \in \chi^* \cup \kappa', j \neq i} d_j^{\gamma_i} \right)$$

$$= g_1^{(i)} \left\{ d_0 \left(\prod_{j \in \chi^* \cup \kappa'} d_j \right) \right\}^{\gamma_i} \left\{ f_0 \left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}}[j]} \right) \right\}^{\gamma_{i,s_{t_i}}} \quad (12)$$

$$\beta_0 = \prod_{i \in V} \left(\beta_{i,0}^*\right)^{\Delta_{i,s}(0)}$$

$$= g_1^{x} \left\{ d_0 \left(\prod_{j \in \chi^* \cup \kappa'} d_j \right) \right\}^{\gamma^*} \left\{ f_0 \left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}}[j]} \right) \right\}^{\gamma'} \quad (13)$$

$$\beta_1 = \prod_{i \in V} \left(\beta_{i,1}\right)^{\Delta_{i,V}(0)} = g_1^{\gamma'} \quad (14)$$

$$u' = \prod_{i \in V} (u_i)^{\Delta_{i,V}(0)} = g_1^{\gamma^*} \quad (15)$$

Where$\gamma^* = \sum_{i \in V} \gamma_i \Delta_{i,V}(0)$
and $\gamma' = \sum_{i \in V} \gamma_{i,s_{t_i}} \Delta_{i,V}(0)$

(d) The signer randomly pics $\delta_0, \alpha, \phi, \theta \in Z_p$& calculates

$$Y_0 = y_0 (h')^{\delta_0 \cdot F(I_d)}$$
$$Y_1 = y_1 . g_1^{\delta_0}$$
$$Y_2 = y_2 \cdot e\left((h')^{\delta_0 \cdot F(I_d)}, g_1\right)$$
$$\nu_1 = \beta_0 \left\{ d_0 \left(\prod_{j \in \chi^* \cup \kappa'} d_j \right) \right\}^{\alpha} \left\{ f_0 \prod_{j=1}^{k} f_j^{b s_{t_i}[j]} \right\}^{\phi} \left(h' \prod_{j=1}^{n} h_j^{m_j} \right)^{\theta} Y_0$$

$$\nu_2 = \beta_1 \cdot g_1^{\phi},\ \nu_3 = u' \cdot g_1^{\alpha},\ \nu_4 = g_1^{\theta}$$

(e) Finally, the signer results in the signature

$$\nu = (Y_1, Y_2, \nu_1, \nu_2, \nu_3, \nu_4).$$

Transform: When the verifier receives the signature $\nu = (Y_1, Y_2, \nu_1, \nu_2, \nu_3, \nu_4)$ of message M, then randomly chooses $a \in Z_p^*$, where $p = |G_1|$. Let $|\kappa^* \cup \kappa'| = n$, the verifier randomly selects a $n-1$ degree polynomial $f(y)$ and $f(0) = a$. He chooses a special element $\rho \in \kappa^* \cup \widehat{\kappa'}$, keeps it secretly and calculates

$\widehat{\nu_1}$

$$= \prod_{i \in V, i \neq \rho} g_1^{f(i)\Delta_{i,\nu}(0)} \left(d_0 \prod_{j \in \chi^* \cup \kappa'} d_j \right)^{\gamma^*} \left\{ f_0 \left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}}[j]} \right) \right\}^{\gamma'} \nu_1$$

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

(16)

$$\widehat{v_2} = g_1^{\gamma'}.v_2,\ \widehat{v_3} = g_1^{\gamma^*}.v_3,\ \widehat{v_4} = v_4$$

(17)

The altered signature $\hat{v} = (Y_1, Y_2, \widehat{v_1}, \widehat{v_2}, \widehat{v_3}, \widehat{v_4})$, together with the message M, is finally sent to the server by the verifier.

Server-aided verify: The server computes

$$\frac{e(\widehat{v_1} g_1)}{e\left\{\widehat{v_2}, f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right\} e\left\{\widehat{v_3}, d_0\left(\prod_{j\in\chi^*\cup\kappa'} d_j\right)\right\} e\left\{\widehat{v_4}, h'\left(\prod_{j=1}^{n} h_j^{m_j}\right)\right\} Y_2} = \Lambda$$

(18)

and gives it back to the verifier.

Lightweight-verify: The verifier computes

$$R = e(g_1, g_1)^a,\ U = e\left(g^{f(\rho)\Delta_{i,v}(0)},\ g_1\right)$$

(19)

and checks whether $\Lambda \cdot U = R \cdot Z$ holds. If it holds, then the signature is valid otherwise the signature is invalid.

Trace: When a signer abuses the signing behaviour, At_{Aut} can track the signer's true identity. The Trace algorithm receives the following inputs: the signature $\nu = (Y_1, Y_2, \nu_1, \nu_2, \nu_3, \nu_4)$, the message $M = (m_1 m_2 \dots m_n) \in \{0,1\}^n$, public parameter params upon the message M concerning the tracing key $tk = y_3$ and the predicate $\gamma_r, \chi^*(\cdot)$ at time period t_i. Based on each possible identity Id, the system determines

$$e\left((h')^{F(I_d)} y_3 . \frac{Y_1}{y_1}\right) = \frac{e(v_1, g_1)}{e\left\{v_2,\ f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right\}} \times \frac{1}{e\left\{v_3,\ d_0\left(\prod_{j\in\chi^*\cup\kappa'} d_j\right)\right\} e\left\{v_4,\ h'\left(\prod_{j=1}^{n} h_j^{m_j}\right)\right\} e(g_1, g_1)^x}$$

If the previously described equation is true, the method returns the signer's actual identity I_d.

VI. SECURITY ANALYSIS

6.1 Correctness

To verify

$$\Lambda \cdot U = R \cdot Z$$

Now,

$$\Lambda \cdot U = \frac{e(\widehat{v_1}, g_1)}{e\left\{\widehat{v_2},\ f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right\} e\left\{\widehat{v_3},\ d_0\left(\prod_{j\in\chi^*\cup\kappa'} d_j\right)\right\} e\left\{\widehat{v_4},\ h'\left(\prod_{j=1}^{n} h_j^{m_j}\right)\right\} Y_2} \cdot e\left(g_1^{f(\rho)\Delta_{i,v}(0)}, g_1\right)$$

$$= \frac{e\left(\prod_{i\in V} \boldsymbol{g_1^{f(i)\Delta_{i,V}(0)}} . v_1, g_1\right)}{e\left(g_1^{\gamma'} v_2, f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right) e\left(g_1^{\gamma^*} v_3, d_0 \prod_{j\in\chi^*\cup\kappa'} d_j\right) e\left(v_4, h' \prod_{j=1}^{n} h_j^{m_j}\right) Y_2}$$

$$= \frac{e\left(\prod_{i\in V} \boldsymbol{g_1^{f(i)\Delta_{i,V}(0)}} . \beta_0 (d_0 \prod_{j\in\chi^*\cup\kappa'} d_j)^{\alpha} \left(f_0 \prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)^{\phi} \left(h' \prod_{j=1}^{n} h_j^{m_j}\right)^{\theta} Y_0, g_1\right)}{e\left(\beta_1 g_1^{\phi}, f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right) e\left(\mathrm{u}' g_1^{\alpha}, d_0 \prod_{j\in\chi^*\cup\kappa'} d_j\right) e\left(g_1^{\theta}, h' \prod_{j=1}^{n} h_j^{m_j}\right) Y_2}$$

$$= \frac{\boldsymbol{e}\left(\prod_{i\in V} \boldsymbol{g_1^{f(i)\Delta_{i,V}(0)}} . \beta_0 Y_0, g_1\right)}{e\left(\beta_1, f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right) e\left(u', d_0 \prod_{j\in\chi^*\cup\kappa'} d_j\right) Y_2}$$

$$= \frac{e\left(\prod_{i\in V} \boldsymbol{g_1^{f(i)\Delta_{i,V}(0)}} . g_1^{x} (d_0 \prod_{j\in\chi^*\cup\kappa'} d_j)^{\gamma^*} \left\{f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right\}^{\gamma'} y_0 (h')^{\delta_0 \cdot F(I_d)}, g_1\right)}{e\left(g_1^{\gamma'}, f_0\left(\prod_{j=1}^{\mathrm{k}} f_j^{b_{s_{t_i}[j]}}\right)\right) e\left(g_1^{\gamma^*}, d_0 \prod_{j\in\chi^*\cup\kappa'} d_j\right) y_2 . e((h')^{\delta_0 \cdot F(I_d)}, g_1)}$$

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

$$= \frac{e\left(\prod_{i\in V} g_1^{f(i)\Delta_{i,V}(0)} \cdot g_1^{x} y_0, g_1\right)}{y_2}$$

$$= \frac{e\left(\prod_{i\in V} g_1^{f(i)\Delta_{i,V}(0)} \cdot g_1^{x}(h')^{\mu_0} \cdot F(I_d), g_1\right)}{e((h')^{\mu_0} \cdot F(I_d), g_1)}$$

$$= \left(\prod_{i\in V} g_1^{f(i)\Delta_{i,V}(0)} \cdot g_1^{x}, g_1\right)$$

$$= \left(g_1^{\sum_{i\epsilon V} f(i)\Delta_{i,V}(0)} \cdot g_1^{x}, g_1\right)$$

$$= e(g_1^{a} . g_1^{x}, g_1)$$

$$= e(g_1, g_1)^{a} . e(g_1, g_1)^{x}$$

$$= R . Z$$

6.2 Traceability

Theorem: Our proposed signature scheme satisfies traceability. Proof: We have

$$\frac{e(v_1,\, g_1)}{e\left\{v_2,\, f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}[j]}}\right)\right\} e\left\{v_3,\, d_0\left(\prod_{j\in\chi^*\cup\kappa'} d_j\right)\right\}} \times \frac{1}{e\left\{v_4,\, h'\left(\prod_{j=1}^{n} h_j^{m_j}\right)\right\} e(g_1, g_1)^{x}}$$

$$= Y_2$$

$$= y_2 . e\left((h')^{\delta_0 . F(I_d)}, g_1\right)$$

$$= e\left((h')^{\mu_0 . F(I_d)}, g_1\right) . e\left((h')^{\delta_0 . F(I_d)}, g_1\right)$$

$$= e\left((h')^{(\delta_0 + \mu_0) . F(I_d)}, g_1\right)$$

$$= e\left((h')^{F(I_d)}, g_1^{(\delta_0 + \mu_0)}\right)$$

$$= e\left((h')^{F(I_d)}, x_3 . g_1^{\delta_0}\right)$$

$$= e\left((h')^{F(I_d)}, x_3 . \frac{Y_1}{y_1}\right)$$

As a result, the aforementioned equation enables the Trace algorithm used by At_{Aut} to reveal the signer's true identity for every potential identification I_d.

6.3 Server Aided Verify

If the server is unreliable, it might try to manipulate the server-aided verification to deceive the verifier into accepting an incorrect signature as valid. However, this is not possible with our protocol. In our scheme, the verifier first selects a random value and keeps them secret, making it impossible for the server to extract α or from β. As a result, the server cannot validate an incorrect signature. Even if the server collaborates with an attacker, it only receives the transformed signature from the verifier and the original signature from the attacker. Despite this, the server cannot determine due to the absence of certain crucial information ρ. Therefore, our server-aided verification method remains fully secure within our signature scheme.

6.4 Attribute Based Privacy

Theorem: The signer's privacy is satisfied by the proposed method.

Proof: The signature in the proposed system does not reveal the signer's attribute set χ, which is utilized to support message M. When the signer's attribute set χ meets the signing predicate $\gamma_{k,\chi^*}(\cdot)$, the signer will create a signature. When χ satisfies the requirement, all we have to do is demonstrate that the given scheme protects the privacy of the signer.

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 9, Issue 10, ISSN No. 2455-2143, Pages 18-29
Published Online February 2025 in IJEAST (http://www.ijeast.com)

Setup: After selecting security parameter λ, C runs the Setup algorithm to generate the master secret key MSK and the public parameter

$$params = (W, \kappa, R_1, R_2, g_1, e, Z, H, D, F)$$

Where,

$$H = (h', h_1, \dots, h_n),\ D = \left(d_0, d_1, \dots, d_{\xi+t-1}\right),$$
$$F = (f_0, f_1, f_2, \dots, f_k)$$

The master secret key MSK and parameter params are returned to A by challenger C.

Queries: After extracting two attribute sets, χ_0 and χ_1, that meet $\gamma_{k,\chi^*}(\cdot)$, A queries KeyGen Oracle. After executing the KeyGen method, challenger C gets signing secret keys in the form of $sk_{t_{i_\vartheta}} = \{y_{0_\vartheta}, y_{1_\vartheta}, y_{2_\vartheta}, sk_{i,s_\vartheta}\}$, where $\vartheta \in \{0,1\}$. Challenger C selects $\mu_{0_\vartheta}, h_{0_\vartheta}, \gamma_{i_\vartheta}, \gamma_{i,s_\vartheta} \in Z_p$ at random and calculates

$$y_{0_\vartheta} = (h')^{\mu_{0_\vartheta}} \cdot H(I_d), y_{1_\vartheta} = g_1^{h_{0_\vartheta}},$$

$$y_{2_\vartheta} = e\left((h')^{\mu_{0_\vartheta} \cdot H(I_d)},\ g_1\right)$$

$$sk_{i,s_\vartheta} = \left\{g_1^{\gamma_{i_\vartheta}}, d_j^{\gamma_{i_\vartheta}}, \left(g_1^{q(i)}(d_0 d_i)^{\gamma_{i_\vartheta}}\right)\right\} \cdot \left\{f_0\left(\prod_{j=1}^{|b_s|} f_j^{b_s[j]}\right)^{\gamma_{i,s_\vartheta}}, g_1^{\gamma_{i,s_\vartheta}}, f_k^{\gamma_{i,s_\vartheta}}\right\}$$

Subsequently, challenger C transmits to A $sk_{t_{i\vartheta}} = \{y_{0_\vartheta}, y_{1_\vartheta}, y_{2_\vartheta}, sk_{i,s_\vartheta}\}$.

Challenge: In order to get an attribute, set meeting $F_{k,\chi^*}(\cdot)$ from either $sk_{t_{i_0}}$ or $sk_{t_{i_1}}$, the adversary A signs an Oracle query on message M^*. C runs the Sign algorithm, randomly chooses $\vartheta \in \{0,1\}$, computes the signature v^* for the signing secret key $sk_{t_{i_\vartheta}} = \{y_{0_\vartheta}, y_{1_\vartheta}, y_{2_\vartheta}, sk_{i,s_\vartheta}\}$ and produces a signature

$$v^* = (Y_1, Y_2, v_1, v_2, v_3, v_4)$$
$$= g_1^{h_0+\delta_0}, e\left((h')^{\delta_0 \cdot F(I_d)}, g_1\right), \left\{d_0\left(\prod_{j \in \chi^* \cup \kappa'} d_j\right)\right\}^{\alpha} \times \left\{f_0\left(\prod_{j=1}^{k} f_j^{b_{s_{t_i}}[j]}\right)\right\}^{\phi} \left\{h'\left(\prod_{j=1}^{n} h_j^{m_j}\right)\right\}^{\theta}, g_1^{\phi}, g_1^{\alpha}, g_1^{\theta}$$

where μ_0, α, ϕ and θ are picked randomly from Z_p.

Guess: A yield an estimate $\vartheta \in \{0,1\}$ about ϑ. The signature generated by either $sk_{t_{i_0}}$ or $sk_{t_{i_1}}$ has an accordant distribution since μ_0, α, ϕ and θ are randomly selected. Therefore, it has been demonstrated that the signature generated by $sk_{t_{i_0}}$ along with attribute set χ_0 may likewise be generated by $sk_{t_{i_1}}$ with attribute set χ_1. The signer's privacy is thus satisfied by the proposed method.

6.5 Properties Comparisons

We compare our presented scheme in Table 2 with some existing signature schemes [9], [5], [4] and [11] with respect to the access policy, server aided verification property, forward security, traceability feature, signature size, and the number of pairings used in the verification process.

Table 2: Comparison Chart

Paper	Access Policy	SAV	SAV -Sec	Forw-Sec	Trac.	Sign. Size	Sign. Ver.
[9]	monotone	×	×	×	✓	$(k + t_{max} + 2\|R_1\| + \|R_2\|$	$(k+5)P$
[5]	LSSS	×	×	×	✓	$(k+3)\|R_1\|$	$(k+3)P$
[4]	tree	✓	✓	×	×	$(2\,no.\,of\,att + 2)\|R_1\|$	$2P + EXP$
[11]	threshold	×	×	✓	✓	$5\|R_1\| + \|R_2\|$	$4P$
Our Sch.	threshold	✓	✓	✓	✓	$5\|R_1\| + \|R_2\|$	$2P$

VII. CONCLUSION

In conclusion, our proposed signature scheme addresses the significant computational burden imposed by existing attribute-based signature (ABS) schemes, which rely heavily on expensive pairing operations. By introducing a traceable and forward-secure attribute-based server-aided verification signature scheme, we have effectively shifted a substantial portion of the computational workload to an external cloud server. This innovative approach not only accelerates the verification process but also alleviates the computational burden on users, making the scheme particularly suitable for those with limited resources. Additionally, our scheme offers a more adaptable access policy, enhancing its practicality and flexibility for various applications. The results demonstrate the potential of server aided verification in optimizing ABS schemes, paving the way for more efficient and user-friendly cryptographic solutions.

Future work will focus on further reducing the computational cost on the verifier's side, specifically by optimizing the remaining exponentiation operations. Additionally, exploring more efficient cryptographic techniques and protocols to enhance overall performance will be a key area of investigation.

VIII. ACKNOWLEDGEMENT

This work is supported by the grant from the State Government of Uttar Pradesh, India sanctioned under Government order no.-47/2021/606/sattar-4-2021-4(56)/ 2020 dated 30/03/2021.

IX. REFERENCE

[1] Boneh, D., Boyen, X., and Goh, E.-J. Hierarchical identiy based encryption with constant size ciphertext. In Annual international conference on the theory and applications of cryptographic techniques (2005), Springer, pp. 440–456.

[2] Canetti, R., Halevi, S., and Katz, J. A forward-secure public-key encryption scheme. In Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22 (2003), Springer, pp. 255–271.

[3] Chen, X., Li, J., Huang, X., Li, J., Xiang, Y., and Wong, D. S. Secure outsourced attribute-based signatures. IEEE transactions on parallel and distributed systems 25, 12 (2014), 3285–3294.

[4] Chen, Y., Li, J., Liu, C., Han, J., Zhang, Y., and Yi, P. Efficient attribute based server-aided verification signature. IEEE Transactions on Services Computing 15, 6 (2021), 3224–3232.

[5] Ding, S., Zhao, Y., and Liu, Y. Efficient traceable attribute-based signature. In 2014 IEEE 13th international conference on trust, security and privacy in computing and communications (2014), IEEE, pp. 582–589.

[6] El Kaafarani, A., Ghadafi, E., and Khader, D. Decentralized trace able attribute-based signatures. In Topics in Cryptology–CT-RSA 2014: The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings (2014), Springer, pp. 327–348.

[7] Escala, A., Herranz, J., and Morillo, P. Revocable attribute-based signatures with adaptive security in the standard model. In Progress in Cryptology–AFRICACRYPT 2011: 4th International Conference on Cryptol ogy in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings 4 (2011), Springer, pp. 224–241.

[8] Girault, M., and Lefranc, D. Server-aided verification: Theory and practice. In Advances in Cryptology-ASIACRYPT 2005: 11th International Conference on the Theory and Application of Cryptology and Information Se curity, Chennai, India, December 4-8, 2005. Proceedings 11 (2005), Springer, pp. 605–623.

[9] Gu, K., Wang, K., and Yang, L. Traceable attribute-based signature. Journal of Information Security and Applications 49 (2019), 102400.

[10] Jakobsson, M., and Wetzel, S. Secure server-aided signature genera tion. In Public Key Cryptography: 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001 Cheju Island, Korea, February 13–15, 2001 Proceedings 4 (2001), Springer, pp. 383–401.

[11] Kang, Z., Li, J., Shen, J., Han, J., Zuo, Y., and Zhang, Y. Tfs-abs: Traceable and forward-secure attribute-based signature scheme with constant size. IEEE Transactions on Knowledge and Data Engineering 35, 9 (2023), 9514–9530.

[12] Li, J., Au, M. H., Susilo, W., Xie, D., and Ren, K. Attribute-based signature and its applications. In Proceedings of the 5th ACM symposium on information, computer and communications security (2010), pp. 60–69.

[13] Li, J., Yan, H., and Zhang, Y. Certificateless public integrity check ing of group shared data on cloud storage. IEEE Transactions on Services Computing 14, 1 (2018), 71–81.

[14] Li, J., Yan, H., and Zhang, Y. Efficient identity-based provable multi copy data possession in multi-cloud storage. IEEE Transactions on Cloud Computing 10, 1 (2019), 356–365.

[15] Lu, Y., Wang, X., Hu, C., Li, H., and Huo, Y. A traceable threshold attribute-based signcryption for mhealthcare social network. International Journal of Sensor Networks 26, 1 (2018), 43–53.

[16] Maji, H., Prabhakaran, M., and Rosulek, M. Attribute based signa tures: Achieving attribute privacy and

collusion-resistance. 2008. EPRINT http://eprint. iacr. org/2008/328 (2008).

[17] Maji, H. K., Prabhakaran, M., and Rosulek, M. Attribute-based signatures. In Cryptographers' track at the RSA conference (2011), Springer, pp. 376–392.

[18] Shanqing, G., and Yingpei, Z. Attribute-based signature scheme. In 2008 International Conference on Information Security and Assurance (ISA 2008) (2008), IEEE, pp. 509–511.

[19] Wei, J., Liu, W., and Hu, X. Forward-secure threshold attribute-based signature scheme. The Computer Journal 58, 10 (2015), 2492–2506.

[20] Wu, W., Mu, Y., Susilo, W., and Huang, X. Provably secure server aided verification signatures. Computers & Mathematics with Applications 61, 7 (2011), 1705–1723.

[21] Yan, H., Li, J., Han, J., and Zhang, Y. A novel efficient remote data possession checking protocol in cloud storage. IEEE Transactions on Infor mation Forensics and Security 12, 1 (2016), 78–88.

[22] Yan, H., Li, J., and Zhang, Y. Remote data checking with a designated verifier in cloud storage. IEEE Systems Journal 14, 2 (2019), 1788–1797.

[23] Yuen, T. H., Liu, J. K., Huang, X., Au, M. H., Susilo, W., and Zhou, J. Forward secure attribute-based signatures. In Information and Communications Security: 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings 14 (2012), Springer, pp. 167 177.

[24] Zhang, Y., Feng, D., Zhang, Z., and Zhang, L. On the security of an efficient attribute-based signature. In International Conference on Network and System Security (2013), Springer, pp. 381–392.