ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

Performance-Optimized Group Signature Schemes Based on the Discrete Logarithm Problem and Bilinear Maps

Kundan Kumar Gupta[1]; Manisha Priya[2]; Aakash Kumar[3]

[1,2]Research Scholar, Department of Mathematics, Vikrant University, Gwalior, M.P.
[3]Research Scholar, Department of Mathematics, Patna University, Patna, Bihar

Publication Date: 2026/09/08

Abstract: Group signature schemes are essential cryptographic primitives that enable authorized members to sign messages on behalf of a group while maintaining anonymity against verifiers. However, many existing schemes face challenges regarding computational efficiency, public key scalability, and robust security against coalition attacks. This study addresses these limitations by proposing three novel group signature constructions. The first scheme is based on the Discrete Logarithm Problem (DLP) and focuses on providing high-speed verification. The second scheme introduces an identity-based (ID-based) structure utilizing both factoring and DLP to simplify key management. The third scheme leverages the algebraic properties of bilinear pairings to create an ID-based group signature that achieves short signature lengths and resists key escrow issues. Formal security analysis demonstrates that the proposed schemes satisfy critical requirements, including unforgetability, anonymity, linkability, traceability, and coalition resistance. Furthermore, a performance comparison indicates that the proposed structures significantly reduce computational complexity in signature generation and verification phases compared to established models like the Lee and Wang-Fu schemes. These findings suggest that the integrated approach of using DLP and bilinear pairings offers a scalable and secure framework suitable for modern digital infrastructure, including e-voting and anonymous electronic tendering systems.

Keywords*: Bilinear Pairings, Cryptography, Discrete Logarithm Problem, Group Signature, Signature Scheme.*

How to Cite: Kundan Kumar Gupta; Manisha Priya; Aakash Kumar (2026) Performance-Optimized Group Signature Schemes Based on the Discrete Logarithm Problem and Bilinear Maps. *International Journal of Innovative Science and Research Technology*, 11(8), 3080-3088. https://doi.org/10.38124/ijisrt/26aug1487

I. INTRODUCTION

➢ *Group Signature*

A group signature scheme allow members of a group to make signature on behalf of the group in such a way that:

- Messages may only be signed by members of the specified group.
- Despite the fact that the receiver of the signature is able to confirm that it is a legitimate signature from that group, they are unable to determine which member of the group signed it.
- In the event that there is a disagreement, the signature may be unsealed by either the members of the group when they are all present or by a group authority in order to expose the identity of the person who signed the message.

- *Properties used in Group Signature Scheme*

A solid group signature scheme should cover a few key points:

Unforgetability means that only group members can sign messages for the group. No outsiders get that power. Anonymity hides who actually signed a message. Besides the group manager, no one can figure out which member it was. Unlinkability makes it tough for anyone to tell if two different signatures came from the same person in the group. Exculpability protects innocent members even if the group manager teams up with bad actors in the group, they still can't forge signatures from members who aren't involved. Traceability gives the manager the ability to reveal which group member created any valid signature.

- ✓ Distinguishability: Every group member has a different private key, so you can tell them apart just by looking at their keys. Non-repudiation: Each private key is unique, and only the owner has access. Once someone signs something, they can't turn around and say they didn't do it.
- ✓ Revocability: The group manager controls membership. Once they revoke someone, that person can't create valid group signatures anymore simple as that. Coalition-resistance: Even if a bunch of members team up and try to

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

cheat the system, they can't make a signature that hides their tracks.

✓ Soundness and Completeness: If a group member creates a valid signature, anyone can check and see that it's legit. If a signature's fake, it won't pass verification no wiggle room. Efficiency: The whole setup works well if the group public key and signature sizes are reasonable, and the algorithms and protocols don't slow things down.

➢ *Procedure for Group Signature Schemes*

A group signature scheme works with these main steps:

- Setup: Give the algorithm a security key, and it spits out a group public key (Y) and a secret key (S) for the group manager.
- Join: Here, the group manager and a user talk back and forth so the user becomes an official member. The user ends up with a membership certificate and a secret.
- Sign: When someone wants to sign a message, they use the group public key, their membership certificate, their secret, and the message itself. The result? A group signature for that message.
- Verify: This step checks if the signature is legit. Just plug in the group public key, the signature, and the original message. The algorithm returns 1 if it's valid, or 0 if it isn't.

➢ *Participants in Group Signature Scheme*

This setup involves three main roles:

- Group Manager: The group manager handles who's in the group and hands out membership keys to members (the signers). They also have the power to let members sign things for the group. If there's ever a dispute, the group manager can reveal who actually signed a particular document first.
- Group Member: A group member holds their own membership key. With this key, they can sign messages on behalf of the group.
- Verifier: This is anyone—maybe the person receiving the group's signature—who wants to check if the group signature is genuine. They can do this with the group's public key.
- Open: This is a specific algorithm. You feed it the message, the signature, and the group manager's secret key. It tells you who signed the message, or if it can't, it just returns "failure."

➢ *Keys in Group Signature Scheme*

This scheme uses three types of keys:

- Master Public Key: Anyone with this key can check whether a group member signed a message.
- Master Secret Key: Every group member has this key so they can sign messages.
- Administrative Key: Only the manager has this one; it lets them figure out which group member actually signed a message.

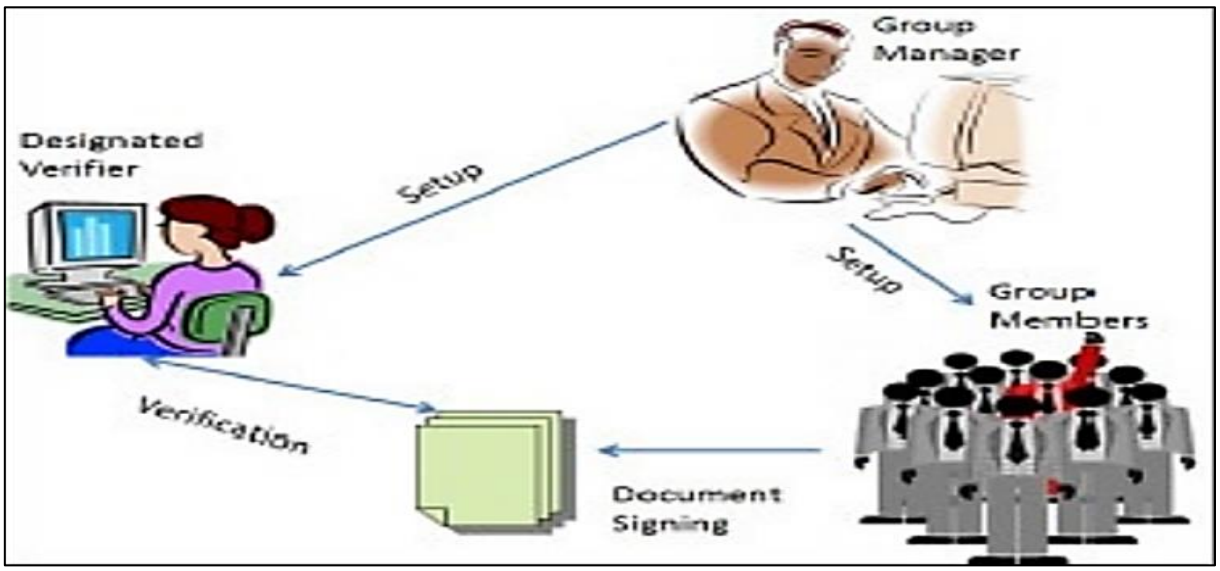

Fig 1 Layout of Standard Group Signature System

➢ *ID-Based Group Signature Scheme*

Park, Kim, and Won introduced a group signature system that relies on identification, but it had a major flaw: the size of both the group's public key and the signature grew with the number of members. To make it work, they had to include each member's identity in the group public key pretty inefficient.

Now, building an ID-based group signature scheme using bilinear pairings is still an open problem. Here's why it's so tough: First off, key escrow is a big drawback in ID-based systems. The trusted authority, called the Key Generation Center (KGC), knows everyone's private key. If someone at the KGC isn't honest, they can forge any member's signature without getting caught.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

Then there's anonymity. A user's public key ID shouldn't reveal anything about who they are. If it does, the whole point of group signatures keeping signers anonymous is lost. Using a random string as a public key brings its own headaches. For one, the KGC could pin a valid group signature on an innocent member, framing them. On the flip side, a member could deny their own signature because the KGC can invent both a public and private key pair for that same random string. Since no one really knows who created a particular public key, there's no clear way to assign responsibility. Group signature scheme can be used in electronic voting, commonly known as e-voting, is a mechanism that allows voters to cast their ballots and digitally count the results of those votes. Electoral authorities and representatives of the government are responsible for the impartial supervision of electronic voting. When it comes to this situation, group signature might be quite helpful. Votes are cast by the voter, and the government authorities are responsible for monitoring and verifying the accuracy of the whole voting process. In situations like this, the best approach is to use group signature. You may use them in electronic tenders. Each and every company that submits a tender form uses the group signature to sign their tender in a manner that is completely anonymous. Immediately after the preferred proposal has been chosen, the lowest bidder may be identified, while the other bidders continue to remain anonymous. For the purpose of monitoring the use of a shared printer inside a network, group signature technique may also be utilised. An organization has a number of computers, all of which are linked to the local network. There is a printer for each department in that firm, and only employees of that department are permitted to use the printer that belongs to their department. Additionally, each department is linked to the network. Because of this, it is necessary to confirm that the user is employed in that particular department before printing that document. However, the firm is concerned about protecting the user's privacy, thus the user's name will not be disclosed. If, on the other hand, at the end of the day, someone learns that a printer has been used excessively, the director has to be able to determine who was responsible for the inappropriate use of that printer.

We suggest four different plans. Regarding the confidentiality of information, there is only one system that corresponds to it. In certain of their schemes, a group manager is not permitted to add members to the group after the initial setup has been completed. The schemes in question are not adaptable and do not relate to the needs that are really being met. Moreover, in the event that there is a disagreement with any of their plans, the group manager is required to get in touch with every member in order to open a signature. There is no efficiency in the plans, and they do not relate to the real requirement that exists.

Back in 1994, two new plans showed up that both let you add new members to a group after the system's first setup. But they handled anonymity differently one followed the stricter standard from information theory, while the other settled for computational anonymity. There was a major flaw though: the group manager could falsely accuse any member of signing a message. In other words, the manager had the power to fake a signature. Fast-forward to 1997, and someone introduced a more effective group signature technique. Like before, it allowed new members to join after setup, and it protected anonymity during the signing process. What's more, they split the group manager's role among several people to balance the workload. But this method still had a big problem. The public key and the signature length got bigger as more people joined, which made things slow and expensive when the group grew large.

Later that same year, a solution appeared. Now, the signature size, public key, and the work needed to sign and verify things all stayed the same, no matter how many members joined. Even if you added more people, you didn't have to change the public key. It was a huge improvement for efficiency. This new system also meant that even if several legitimate members teamed up to produce a group signature, the manager still couldn't figure out who actually signed it. But then in 1999, someone found a flaw. In rare cases, members working together could create a group signature that hid the signer's identity completely from the manager. Still, people only saw this attack in special situations. They introduced an improved plan back in 2000. It made the group's public key size and signature length stay the same, no matter how many people were in the group. This approach stuck to security standards, too. Honestly, what Camenisch came up with was faster and safer than earlier methods. Interestingly, you don't even need a deletion function if someone leaves the group. Back in 1997, the author worked on an ID-based group signature system. But the problem was, both the public key and the signatures grew longer as the group got bigger. Plus, tests showed the plan didn't really protect anonymity. Then in 1998, someone proposed a group signature method based on the discrete logarithm problem. Still, several researchers pointed out serious security flaws in their technique. In (1999) they made an effort to find a solution to the issue. They suggested a revolutionary group signature system that was based on identification. However, it is not immune to coalitions and may be formed by anybody in the world. Additionally, in the year 2002, he made Lee better and brought to light the fact that Shi's strategy is likewise vulnerable. Within the same year, a group signature scheme that had a high degree of separability was suggested. In other words, the revocation manager is able to do their duties independently of the membership manager participation. It was pointed out in 2003 that the system proposed by Xia and You does not meet the requirements of unlinkability and unforgeability. Within the same year, the author put out a proposal for a safe group signature structure. It was pointed out by Cao and colleagues that Wang and Fu's scheme cannot be traced.

A novel ID-based group signature method that is derived from bilinear pairings was suggested by the author. The technique offered a solution to the issue of key escrow using a particular approach. In their plan, a trusted third party in the ID-based system, known as KGC, may be dishonest. If KGC impersonates an honest user in order to sign a document, the user has the ability to offer evidence that the KGC is dishonest. Additionally, Lee and Chang's plan was enhanced in the year 2010. Since that time, a number of other

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

group signature schemes have been suggested as well as improvements to those schemes.

II. METHODOLOGY

The Initiation phase, the Signing phase, the Verification phase, and the Identification phase are the four stages that may be used to describe our strategy. This step of our suggested plan is the same as the one described in.

➢ *Initiation Phase*

Suppose p and q are two prime numbers of a large size, such that q divides p minus one, or q divides p minus one. For the sake of $GF(p)$, let g be a generator of order q. Every member of the group, denoted as t_i, is assigned a secret key, denoted as x_i, and the public key, denoted as y_i, is calculated by multiplying the secret key with the modulo of p. Let us assume that T is the group authority, possessing a secret key denoted as x_T and a public key denoted as y_T, which is equal to the product of gx_T and p.

For every member of the group, denoted as t_i, the group authority T performs the computation of (r_i, s_i) in such a way that ri is equal to g minus $k_i y_i k_i$ modulo p and s_i is equal to g_i minus $r_i\, x_T$ modulo q, where k_i is a random integer such that gcd(k_i, q) equals 1. After then, the authority T communicates (r_i, s_i) to the member of the group t_i in a covert manner. The correctness of the (r_i, s_i) may be verified by the group member t_i after they have received the (r_i, s_i) information.

$$g^{s_i} y_T^{r_i} r_i mod\ p \equiv \left(g^{s_i} y_T^{r_i}\right)^{x_i} mod\ p \quad (1)$$

- *Signing Phase*

To sign message m, the group member ti first randomly selects three numbers $a, b, t \in Z_q^*$ such that $gcd(a, q) = 1, gcd(b, q) = 1, gcd(t, q) = 1$ and computes (r, A, B, C, D, E) as follows:

$$r = r_i b^{-1} - a^{-1}\ mod\ q \quad (2)$$

$$A = y_T^b mod\ p \quad (3)$$

$$B = ar + 1\ mod\ q \quad (4)$$

$$C = s_i a\ mod\ q \quad (5)$$

$$D = r_i^a mod\ p \quad (6)$$

$$E = y_T^{axi} mod\ p \quad (7)$$

Now we compute

$$W_i = A^B g^C mod\ p \quad (8)$$

$$R = W_i^t mod\ p \quad (9)$$

Then the group member to solve the congruence relation:

$$h(m) = Rx_i + tS\ mod\ q \quad (10)$$

Thus signer generates the group signature on m as (h(m), R, S, A, B, C, D, E) and send it to the receiver.

- *Verification Phase*

After receiving the signature, the verifier computes:

$$W_i = AB_g C\ mod\ p \quad \text{and } DH_i = W_i D\ mod\ p \quad (11)$$

Then he checks the congruence relation:

$$W_i^{h(m)} \equiv R^S (DH_i)^R mod\ p \quad (12)$$

If above result holds, the group signature on m (h(m), R, S, A, B, C, D, E) is valid.

- *Identification Phase*

In the event that there is a disagreement, the group signature is unsealed in order to show the signer's identity. To identify the signer, First, the group authority T has access to the key k_i of each group member t_i. After that, the value of r_i is calculated in such a way that it is equal to the product of g and $k_i y_i k_i$ modulo p, where yi represents the public key of t_i. After that, the authority T will choose a candidate at random, it will verify the congruence relation, and then it will proceed.

$$DH_i = y_i C E r_i\ mod\ p \quad (13)$$

The group authority T will proclaim that t_i is the signer who signed the message if the relation described above turns out to be true; otherwise, they will go on to the next candidate.

- *Security Analysis*

The security of our scheme is evaluated in accordance with the standards that are outlined in (1.1.1), taking into consideration the fact that a secure group signature scheme has to fulfil a number of security features.

Correctness: Verification of (1.1)

$$\begin{aligned} g^{s_i} y_T^{r_i} r_i &= g^{(k_i - r_i x_T)} g^{x_T r_i} r_i \bmod p \\ &= g^{k_i} r_i \\ &= g^{k_i} g^{-k_i} y_i^{k_i} \\ &= g^{x_i k_i} \end{aligned}$$

And

$$\begin{aligned} \left(g^{s_i} y_T^{r_i}\right)^{x_i} &= \left(g^{k_i - r_i x_T} g^{x_T r_i}\right)^{x_i} \bmod p \\ &= g^{k_i x_i} \end{aligned}$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

Verification of (1.4.12)

$$\begin{aligned} W_i &= A^B g^C \bmod p \\ &= (y_T^b)^{(ar+1)} g^{s_i a} \\ &= y_T^{(abr+b)} g^{s_i a} \end{aligned}$$

$$\begin{aligned} &= y_T^{ab[r_i b^{-1} - a^{-1}]+b} g^{s_i a} \\ &= y_T^{(ar_i - b + b)} g^{s_i a} \\ &= y_T^{ar_i} g^{s_i a} \\ &= g^{x_T a r_i} g^{(k_i - r_i x_T)a} \\ &= g^{ak_i} \end{aligned}$$

And

$$\begin{aligned} DH_i &= W_i D \bmod p \\ &= g^{k_i a} r_i^a \\ &= g^{k_i a} g^{-k_i a} y_i^{k_i a} \\ &= g^{k_i x_i a} \end{aligned}$$

The correctness of equation (1.4.12) can be verified as follows:

$$\begin{aligned} W_i^{h(m)} &= W_i^{Rx_i + ts} \bmod p \\ &= W_i^{Rx_i} W_i^{ts} \\ &= (W_i^t)^s W_i^{Rx_i} \\ &= R^S (W_i^{x_i})^R \\ &= R^S (DH_i)^R \end{aligned}$$

Verification of (1.4.13)

$$\begin{aligned} DH_i &= W_i D \\ &= W_i r_i^a \\ &= g^{ak_i} g^{-ak_i} g^{ak_i x_i} \\ &= g^{ak_i x_i} \end{aligned}$$

And

$$\begin{aligned} y_i^C E^{r_i} &= g^{x_i C} y_T^{a x_i r_i} \\ &= g^{x_i s_i a} y_T^{a x_i r_i} \\ &= g^{x_i a (k_i - r_i x_T)} g^{x_T x_i r_i a} \\ &= g^{ak_i x_i} \end{aligned}$$

- Unlinkable: In the event that no one is able to determine whether or not two distinct group signatures were indeed issued by the same group member, then the signatures cannot be linked together. Following the identification of the group signature on m as $(h(m), R, S, A, B, C, D, E)$, the verifier validates the group signature, and he is aware of the signer based on the announcement made by the authority. The group signature for another message m' is $(h(m'), R', S', A', B', C', D', E')$. This is the case if t_i signs another message on m'. It is not possible to find any parameters that are comparable between the expressions $(h(m), R, S, A, B, C, D, E)$ and $(h(m'), R', S', A', B', C', D', E')$.

Consequently, there is no connection between the two signatures that were produced by the same member. Therefore, our plan cannot be linked to anything.

Unforgeable: Only someone inside the group can create a real group signature. In our scheme, we use three parameters a, b, and t from Z_q^*, and they sit in the exponent of the equation. That bumps up the security, and since everything relies on the discrete logarithm problem, it's tough for anyone to crack this. Traceability: The group manager can figure out exactly which user signed something, as long as they get a valid signature. The secret key k_i is only known to the authority. Without that key, nobody else can calculate r_i or check the congruence relation. So the group authority T can trace the signer back every time.

$$\mathrm{DH_i} = \mathrm{y_i^C E^{r_i} mod\ p}$$

Coalition-resistance: In our scheme, colluding members cannot generate s_i without knowing the secret x_T of group authority. Therefore colluding members cannot generate C and W_i. Thus our signature scheme is coalition-resistant.

➢ *Proposed Scheme II*

(ID-Based Group Signature Scheme Based on Factoring and DLP)

System Setup, Extract, Join, Sign, Verify, and Open are the six algorithms that make up the identity-based group signing system. The Trusted authority (TA), the Group authority (GA), the Signer, and the Verifier are the four people that are involved in the system by definition. The Tseng-Jan group signature system serves as the foundation for our approach.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

- *System Setup*

Let $N = pq$ where p and q are very large prime number and $\varphi(N) = (p-1)(q-1)$ is a ϕ Euler function. Let ID_i be a string denoting the identity of a user U_i and h(.) be a publicly known cryptography hash function. We require another hash function $H_1:\{0,1\}\rightarrow Z_N$.

- *Extract*

First user U_i sends his identity information ID_i to trusted authority, then trusted authority selects integers $a, r, k, e, v \in Z_N^{\star}$ such that $ke \equiv 1 \bmod \varphi(N)$ and $rv \equiv 1 \bmod \varphi(N)$, and $QID_i = H_1(ID_i)$ and computes:

$$T = (Q_{IDi})^a \bmod N \quad (14)$$

$$R = g^k T^r \bmod N \quad (15)$$

$$u = g^k \bmod N \quad (16)$$

Where, g of large order in Z_N. Now trusted authority sends (T, R) to the group authority (GA).

- *Join*

Now suppose user U_i wants to join the group. For this, group authority selects integers $x, z \in Z_N$ such that $xz \equiv 1 \bmod \varphi(N)$ and computes.

$$y = g^x \bmod N \quad (17)$$

$$S = R^x \bmod N \quad (18)$$

Thus membership certificate for user U_i is (S, T). The system public parameters are (N, g, y, e, z, u, v) and the secret parameters are (a, k, x, r).

- *Sign*

Now user U_i selects an integers $\alpha, \beta \in Z_N$ and computes.

$$A = T^{\alpha} \bmod N \quad (19)$$

$$B = y^{\beta} \bmod N \quad (20)$$

$$C = S^{\beta eh(m\|A\|B)+\alpha z} \bmod N \quad (21)$$

$$D = u^{\alpha} \bmod N \quad (22)$$

Thus ID-based group signature on the message m is (A, B, C, D).

- *Verify*

First verifier uses system public parameter and checks the following relation.

$$Cv \bmod N \equiv [\beta e \, logT + v]h(mkAkB)A.Dv \bmod N \quad (23)$$

If it holds, then (A, B, C, D) is valid ID-based group signature on the message m.

- *Open*

Finally, in case of dispute, the group authority can open the signature to reveal who signed it by checking the relation.

$$Ax \equiv DxelogT \bmod N. \quad (24)$$

- *Security Analysis*

The security of our scheme is evaluated in accordance with the standards that are outlined in (1.1.1), since a secure group signature scheme ought to be able to fulfil a number of security features.

✓ Correctness: In our scheme, signature provided by group members must be accepted by verifier.

Verification:

$$\begin{aligned} C^v &= (S^{\beta eh(m\|A\|B)+\alpha z})^v \bmod N \\ &= S^{(\beta evh(m\|A\|B)+\alpha zv)} \\ &= R^{(x\beta evh(m\|A\|B)+\alpha zxv)} \\ &= (g^k T^r)^{x\beta evh(m\|A\|B)+\alpha zxv} \\ &= g^{kx\beta evh(m\|A\|B)+\alpha zxkv}.(Q_{IDi})^{rax\beta evh(m\|A\|B)+\alpha zxvar} \\ &= g^{x\beta vh(m\|A\|B)+\alpha kv}.(Q_{IDi})^{ax\beta eh(m\|A\|B)+\alpha a} \end{aligned}$$

Now

$$\begin{aligned} &(B^{elogT+v})^{h(m\|A\|B)}.A.D^v \bmod N \\ &= (g^{x\beta elogT+x\beta v})^{h(m\|A\|B)}.(Q_{IDi})^{a\alpha}.g^{k\alpha v} \bmod N \\ &= g^{x\beta vh(m\|A\|B)+\alpha kv}.(Q_{IDi})^{ax\beta eh(m\|A\|B)+\alpha a} \end{aligned}$$

Verification:

$$Ax = T\alpha x \bmod N = (QIDi)a\alpha x$$

$$\begin{aligned} D^{xelogT} \bmod N &= u^{\alpha xelogT} \bmod N \\ &= g^{k\alpha xelogT} \\ &= (g^{logT})^{\alpha x} \\ &= T^{\alpha x} \\ &= (Q_{IDi})^{a\alpha x} \end{aligned}$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

- ✓ Unforgeability: According to our plan, only members of the group are authorised to sign communications on behalf of the group owing to the fact that the membership certificate for user U_i is (S, T), which is something that is created by trusted authority and group authority.
- ✓ Anonymity: The only person who is allowed to open the signature and disclose who signed it is the group manager, according to our policy owing to the fact that the secret key x is only known by the group manager, and it is computationally difficult to discover out $x = z - 1\ mod\ \varphi(N)$
- ✓ Unlinkability: All of the items are dispersed in a consistent manner inside their respective operation groups when an ID-based group signature is applied to the message m, which is (A, B, C, D). Therefore, it is impossible to differentiate between any of the group signatures. Due to the fact that the members of the group are indistinguishable from one another, their signatures cannot be linked together.
- ✓ Traceability: It is possible for the trusted authority to quickly identify the real signer of a valid signature by examining the public key component in the group signature. This is because the trusted authority generates each member's private key from their public keys in such a way that ke is greater than or equal to one mod N and rv is more than or equal to one mod N.
- ✓ Coalition-resistance: Membership certificates for users are denoted by the letters (S, T) in our system. In the absence of knowledge of the private keys a and x of the trusted authority and the group authority, respectively, members who are conspiring are unable to create membership certificates at the S and T levels. Therefore, members who are intentionally conspiring are unable to calculate A and C of the ID-based group signature technique. As a result, our signature method is immune to coalition politics

➢ *Proposed Scheme III*

(ID-Based Group Signature Scheme from Bilinear Pairings)

An ID-based group signature technique that is derived from bilinear pairing is what we propose. It is sufficient for us to take into account the fact that the Key generation centre (KGC) is the group manager. The plan that has been proposed includes six different steps: setup, extraction, joining, signing, verifying, and opening. In the context of our plan, KGC is no longer considered to be a reliable stakeholder.

Given that G1 is a Gap Diffie-Hellman cyclic additive group formed by P, with an order of prime order q, and G2 is a cyclic multiplicative group of the same order q, let us consider both of these groups. It is a map that is a bilinear pairing.

$e : G1 \times G1 \rightarrow G2$ Define cryptographic hash functions are:

$$H_1 : \{0,1\}^* \times G_1 \rightarrow G_1 \text{ and } H_2 : \{0,1\}^* \rightarrow Z_q^*,\ H_3 : G_1 \rightarrow Z_q^*.$$

- *Setup*

KGC chooses a generator P of G1 and picks a random number $s \in Z_q^*$ and set $Ppub = sP$. Thus system parameters are $\{G1, G2, e, q, P, Ppub, H1, H2, H3\}$ and s is the master secret key, which is known only to him.

- *Extract*

A user U_i submits his (or her) identity information ID_i and rP to KGC, where $r \in Z_q^*$ long-term private key. Then KGC computes the user's public key $QID_i = H1(ID_i\ k\ T, rP)$. Here T is validity period of r and sends $SID_i = sQID_i$ to the user U_i via a secure channel. Thus user's private key pair is (r, SID_i). The user should update his key pair after period T. We call SID_i, rP pseudo secret, since KGC is no longer trustful, it may expose them to other members.

- *Join*

Suppose that a user U_i wants to join the group. For this, he and KGC perform Join protocol as follows:

The user U_i chooses a random number, $x_i \in Z_q^*$ then sends $\{rx_iP, rP, ID_i, x_iP\}$ to KGC.

If KGC is convinced that the user know $D_i = sH1(ID_i\ k\ T, rP)\ and\ e(rx_iP, P) = e^{(x_i, rP)}$, KGC sends secretly $s_i = sH1(ID_i, rx_iP)$ to the user U_i.

Thus user's member certificates are (s_i, rx_iP) and his private signing key is rxi. KGC adds rx_iP, x_iP, rP, ID_i to the member list.

- *Sign*

To sign a message m, the user U_i randomly chooses number $\alpha, \beta, k \in Z_q^*$ and uses his signing key and corresponding member certificate and then computes the following values:

$$R = kP \quad (25)$$

$$S1 = rx_iQID \quad (26)$$

$$S2 = rx_i\alpha H1(ID_i, rx_iP) + H3(R)P \quad (27)$$

$$S3 = rx_i\beta H1(ID_i, rx_iP) - H2(m)P \quad (28)$$

$$S4 = [H2(m)\alpha + \beta H3(R)]s_i \quad (29)$$

Thus ID-Based group signature on the message m is $[R, S1, S2, S3, S4, rx_iP]$

- *Verify*

To verify a group signature [R, S1, S2, S3, S4, rxiP] on the message m. Verifier accepts the signature if following equations hold:

$$\hat{e}(S_4, rx_iP) = \hat{e}(S_2, P_{pub})^{H_2(m)}\hat{e}(S_3, P_{pub})^{H_3(R)} \quad (30)$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

$$\hat{e}(S_1, P) = \hat{e}(Q_{ID_i}, rx_i P) \quad (31)$$

If above equations are true then $[R, S1, S2, S3, S4, rx_i P]$ is valid ID-based group signature on the message m.

- *Open*

In case of dispute, the KGC can easily identify the user. The signer cannot deny the signature after KGC presents a proof. KGC checks the following equations:

$$\hat{e}(S_{ID_i}, P) = \hat{e}(H_1(ID_i \parallel T, rP), P_{pub})$$

$$\hat{e}(s_i, P) = \hat{e}(H_1(ID_i, rx_i P), P_{pub})$$

$$\hat{e}(S_1, P_{pub}) = \hat{e}(S_{ID_i}, rx_i P)$$

And

$$\hat{e}(S_2, S_{ID_i})^{H_2(m)} \hat{e}(S_3, S_{ID_i})^{H_3(R)} = \hat{e}(S_4, S_1) \quad (32)$$

- *Security Analysis*

In this section, we prove the security of our group signature scheme on the assumption that G1 is Gap DH group.

✓ Theorem: If an attacker, (say A1), manages to forge a member certificate with time t and a noticeable chance ε (without teaming up with the KGC), then we can actually use A1 to solve the CDHP in G1 within about the same time t and with the same probability ε. Here's how it goes. A1 forges a legit pseudo secret key and a member certificate again, with some solid chance ε by following a specific strategy. First, A1 hits the random oracle H1 (.) up to t times. Then, A1 comes up with a 5-tuple like (ID_i, rP, rx_i P, SID_i, s_i), and the important bit here is this: the triple (ID, rP, rxP) wasn't something they ever asked the oracle about before. If this 5-tuple turns out to be valid, it has to pass the open function test…

$$\hat{e}(S_{ID_i}, p) = \hat{e}(H_1(ID_i \parallel T, rp), P_{Pub})$$

$$\hat{e}(s_i, P) = \hat{e}(H_1(ID_i, rx_i P), P_{Pub})$$

Let $H1(ID\ k\ rP)\ OR\ H1(ID\ k\ rxP) = aP, Pub = bP$. Then adversary can solve CDHP in G1 for $SID = S = abP$ with negligible probability ε.

✓ Theorem: If we base the security of our proposed ID-based group signature scheme on the difficulty of the CDHP in the random oracle model. Let's break down the proof. A secure group signature scheme has to meet a few key requirements. We're going to walk through each security property and show how our scheme checks out, following the criteria outlined earlier.

Correctness: Verification:

$$\begin{aligned}
&\hat{e}(S_2, P_{pub})^{H_2(m)} . \hat{e}(S_3, P_{pub})^{H_3(R)} \\
&= \hat{e}(rx_i \alpha H_1(ID_i, rx_i P) H_2(m) + H_2(m) H_3(R) P, P_{pub}) . \\
&\quad \hat{e}(rx_i \beta H_1(ID_i, rx_i P) H_3(R) - H_3(R) H_2(m) P, P_{pub}) \\
&= \hat{e}(rx_i \alpha H_1(ID_i, rx_i P) H_2(m) + rx_i \beta H_1(ID_i, rx_i P) H_3(R), \\
&\quad P_{pub}) \\
&= \hat{e}(\alpha H_2(m) s_i + \beta H_3(R) s_i, rx_i P) \\
&= \hat{e}(S_4, rx_i p)
\end{aligned}$$

And

$$\begin{aligned}
\hat{e}(S_1, P) &= \hat{e}(rx_i Q_{ID_i}, P) \\
&= \hat{e}(Q_{ID_i}, rx_i P)
\end{aligned}$$

Verification of (1.8)

$$\begin{aligned}
&\hat{e}(S_2, S_{ID_i})^{H_2(m)} . \hat{e}(S_3, S_{ID_i})^{H_3(R)} \\
&= \hat{e}(rx_i \alpha H_1(ID_i, rx_i P) H_2(m) + H_2(m) H_3(R) P, S_{ID_i}) . \\
&\quad \hat{e}(rx_i \beta H_1(ID_i, rx_i P) H_3(R) - H_3(R) H_2(m) P, S_{ID_i}) \\
&= \hat{e}(rx_i \alpha H_1(ID_i, rx_i P) H_2(m) + rx_i \beta H_1(ID_i, rx_i P) H_3(R), \\
&\quad S_{ID_i}) \\
&= \hat{e}(rx_i \alpha H_2(m) s_i + rx_i \beta H_3(R) s_i, Q_{ID_i}) \\
&= \hat{e}(S_4, S_1)
\end{aligned}$$

✓ Unforgeability: Only actual group members can sign messages on the group's behalf. That's because each user's membership certificate (Si, rx_i P) comes from the KGC.

✓ Anonymity: Only the KGC can figure out who signed a message. The random value xi keeps things private, and $rx_i P$ gives away nothing about who the user is, unless you're the KGC.

✓ Unlinkability: If you see two values like $rx_i P$ and $rx_j P$, it's nearly impossible to tell if they belong to the same person unless you know the corresponding $x_i P$ and $x_j P$. That keeps signatures from being linked to each other.

✓ Exculpability: No group member can sign as someone else, because they don't have access to other members' private keys. Even the KGC, who holds s_i for each user, doesn't know anyone's personal rx_i.

III. PERFORMANCE ANALYSIS

How complex a signature scheme is really comes down to four main operations: exponentiation, multiplication, finding inverses, and hash functions. Here's how to break it down:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26aug1487

These are the building blocks used to analyse how each scheme performs.

- TM: computation time requirement for modulo multiplication.
- TE: computation time requirement for modulo exponentiation.
- TI: computation time requirement for modulo inverse operation.
- TH: one time running of hash operations.

Table 1 Comparison of Proposed Scheme I with Lee's & Wang-Fu Scheme

Phases	Signature Generation		Signature Verification		
C. Chi Lee's Scheme	24TM + 7TE+ 1TH		5TM	+ 7TE	+ 1TH
Wang-Fu's Scheme	13TM	+ 12TE +5TH + 1TI	7TM	+ 9TE	+ 5^{TH}
Proposed Scheme I	7TM	+ 6TE + 1TH+ 2TI	3TM	+ 5TE	+ 1TH

Table 2 Comparison of Proposed Scheme II with Tseng's Scheme

Phases	Signature Generation	Signature Verification		
Tseng-Jan	5TM + 3TE+ 2TH	3TM	+ 4TE	+ 2TH
Proposed Scheme II	3TM + 4TE + 1TH	3TM	+ 4TE	+ 1TH

In this evaluation, the time for performing modulo addition and subtraction operations are ignored.

Based on the tables shown above, we can see that the computing complexity of the proposed scheme I and II is lower than that of the system that is already in place.

IV. CONCLUSION

The schemes we introduce in this study are quite simple, requiring less computational effort. In our plan, the group's public key and the signature length don't grow as the group gets bigger, which keeps things efficient regardless of how many people join. The method stands strong because of reliable hash functions and solid Diffie-Hellman calculations, additionally, it is hard to break due to the complex discrete logarithm challenge. We show that our system covers all the important security features you'd expect from a group signature schemes like unlinkability, unforgeability, traceability, coalition-resistance, and more. Our results even improve on certain previous studies in this field.

REFERENCES

[1]. Agarwal, A., & Saraswat, R. (2013). A survey of group signature technique, its applications and attacks. *International Journal of Engineering and Innovative Technology*, 2(10).

[2]. Ateniese, G., & Tsudik, G. (1999). Some open issues and new directions in group signature schemes. *In Financial Cryptography (FC'99)*, LNCS 1648, 196–211. Springer.

[3]. Ateniese, G., Camenisch, J., Joye, M., & Tsudik, G. (2000). A practical and provably secure coalition-resistant group signature scheme. *In Advances in Cryptology – CRYPTO 2000*, LNCS 1880, 255–270. Springer.

[4]. Camenisch, J. (1997). Efficient and generalized group signatures. *In EUROCRYPT 1997*, LNCS 1233, 465–479. Springer.

[5]. Camenisch, J., & Stadler, M. (1997). Efficient group signature schemes for large groups. *In CRYPTO 1997*, LNCS 1294, 410–424. Springer.

[6]. Camenisch, J., Piveteau, J. M., & Stadler, M. (1994). Blind signatures based on the discrete logarithm problem. *In EUROCRYPT 1994*, 428–432.

[7]. Cao, Z. (2005). Untraceability of two group signature schemes. *Cryptology ePrint Archive*, Report 2005/055.

[8]. Cheng-Chi, L., Chang, T. Y., & Hwang, M. S. (2010). A new group signature scheme based on the discrete logarithm. *Journal of Information Assurance and Security*, 5, 54–57.

[9]. Furukawa, J., & Imai, H. (2006). An efficient group signature scheme from bilinear maps. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(5).

[10]. Joye, M., Kim, S., & Lee, N. (1999). Cryptanalysis of two group signature schemes. *In Information Security 1999*, LNCS 1729, 271–275. Springer.

[11]. Lee, W. B., & Chang, C. C. (1998). Efficient group signature scheme based on discrete logarithm problem. *IEE Proceedings – Computers and Digital Techniques*, 145(1), 15–18.

[12]. Tseng, Y. M., & Jan, J. K. (1998). A novel ID-based group signature. *In Workshop on Cryptology and Information Security*, 159–164.

[13]. Wang, X., & Fu, F. (2003). A secure group signature scheme. *Journal of Electronics and Information*.

[14]. Xia, S., & You, J. (2002). A group signature scheme with strong separability. *The Journal of Systems and Software*, 60(3), 177–182.

[15]. Zhang, J., Wu, Q., & Wang, Y. (2005). A new efficient group signature with forward security. *Informatica*, 29, 321–325.

[16]. Zhou, S., & Lin, D. (2005). On anonymity of group signatures. *In CIS 2005*, LNAI 3802, 131–136. Springer.