Int. j. inf. tecnol. (October 2022) 14(6):2927–2939
https://doi.org/10.1007/s41870-022-01043-3

ORIGINAL RESEARCH

A provably secure data sharing scheme for smart gas distribution grid using fog computing

Rachana Y. Patil[1] · Yogesh H. Patil[2] · Renu Kachhoria[1] · Savita Lonare[1]

Received: 15 April 2022 / Accepted: 22 July 2022 / Published online: 13 August 2022
© The Author(s), under exclusive licence to Bharati Vidyapeeth's Institute of Computer Applications and Management 2022

Abstract The concept of smart gas distribution grid implies to enhancing current gas distribution grids by establishing continuous on request and bidirectional data interchange between metering devices, gas flow instrument's, utilities, and end clients. Smart gas distribution grid (SGDG) integrates IoT in gas distribution system and enhances the management in hazard minimization for gas infrastructure. As a result, the security and privacy of this type of vital infrastructure are frequently overlooked during the design process. Interaction over wireless channel and the lack of computing power on the SGDG make it impossible to use for secure operations. As a solution, we have developed a hyper elliptic curve based proxy signcryption scheme. Data transmission between smart gas metres and a cloud server is supported by a fog layer that provides excellent response times, reliability, and enhanced privacy. Hyper Elliptic Curve Cryptography (HECC) is the foundation of the proposed scheme, which enhances network computation efficiency. Formal security analysis is used to assess the toughness of security measures. Under OFMC and CL-Atse backend, the simulation study using AVISPA tool shows that the proposed scheme is safe. The computation and communication costs of the proposed scheme have also been compared to those of the relevant existing schemes in the performance analysis. The security and performance evaluations show that the proposed scheme is superior.

Keywords Proxy signcryption · Security · Privacy · Smart gas grid · AVISPA · Fog computing

1 Introduction

Due to continuous increase in prices of liquid petroleum gas (LPG) and crudeoil (fossil fuel), natural gas is the best option as fuel with better parameters and has increase in demand for domestic and automobile industry [1, 2]. The processing and distribution of natural gas up to the end users encounters several challenges like leakages in pipe, bursting due to over pressure, contaminations due changing weather conditions and maintaining pressure and flow through pipes.

To deal with these difficulties, access to information related to gas and its grid is very important to develop advanced strategies for gas distribution management. Combining information and Communication technologies (ICT) into the present gas transportation system is one feasible salutation to gather the gas associated data. This upgraded infrastructure that configures ICT into the gas distribution approach [3] is termed as Smart Gas Distribution Grid (SGDG) Fig. 1 describes the overview of Smart Gas Distribution Grid.

By establishing bidirectional data exchange between metering devices, the smart gas distribution grid concept aims to improve existing gas distribution grids. Utilities, gas flow instruments, and end users are all involved. IoT-enabled gas distribution grids provide the same benefits as water and electrical grids while also improving gas infrastructure management in terms of hazard minimization.

Deploying smart metering [4, 5] is believed to be of vital importance towards the fulfilment of smart towns as they support numerous benefits to gas utilities and end clients. In comparison with conventional hand operated metering

✉ Rachana Y. Patil
rachana.patil@pccoepune.org

[1] Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

[2] Dr. D. Y. Patil Institute of Technology, Pune, Maharashtra, India

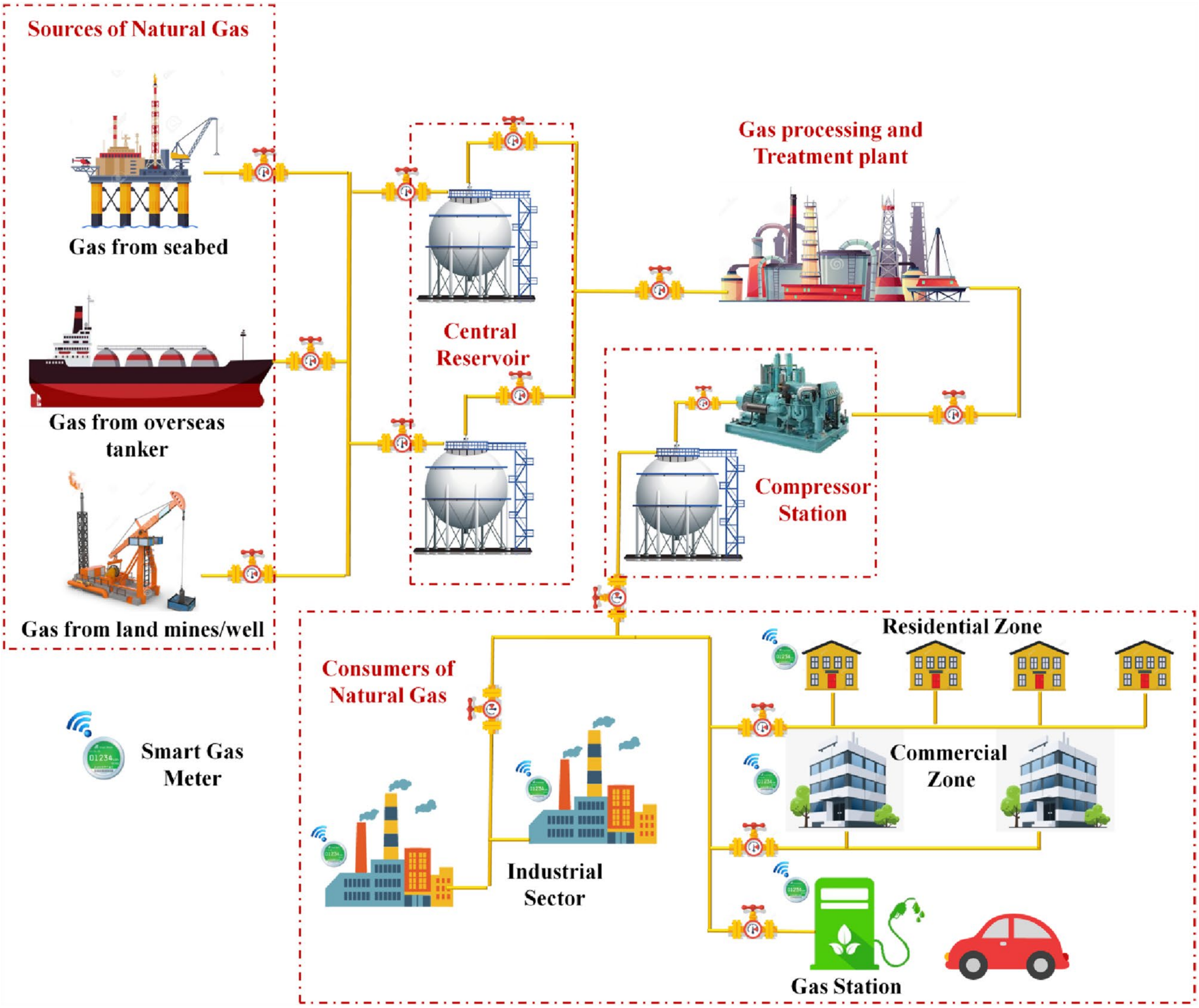


Fig. 1 Overview of smart gas distribution grid (SGDG)

devices smart gas meters provides precise data acquisition during low pressures, moisture contents in gas which can indicates the problems in system and are more resistive to corrosions from grains in the system. The data gathered from subsequent infrastructure supports the analysis of gas demands that supports better understanding of gas utilization, this encourages the designing of urban gas distribution network [6].

The accessibility of instantaneous data at elevated temporal frequency supports the gas services detecting leakages and fixtures malfunctioning, timely scheduled repair or upgrade of the infrastructure and eventually supports them to take on desired goals for natural gas utilization. Such expertise derived from acquired data also helps in connection with gas utilization demand models to recognize the parameters adding to peak demand. Furthermore, innovative optimization methods can be applied to enhance the gas saving through the urban gas life cycle means improving in operational effectiveness for the proprietor of gas delivery network. Looking ahead the use of data recent improvements in smart network actuators help to develop self-sufficient smart grids, where metering and actuators coordinate to manage the delivery network more effectively than any manual service.

From IOT point of view, one major challenge is the vast amount of data created by smart gas meters and the way it communicates effectively through all elements of system [7]. The multi-modal frameworks of town-based utilization and requirement of data access between various groups introduces additional technical difficulties on transferring enormous database acquired across various infrastructures.

The latest advancements of the upcoming 5th generation communication networks (5G) are likely to enhance the implementation of fog computing with lots of benefits with respect to response time, delays in transmission, cost of energy management in time dependent applications.

Hierarchical arrangement of cloud-fog computing supports various forms of computing services that enhances the resource managing in smart grids [8, 9]. The real-world testimonial in deployments reflects the gains of middleware technologies. Furthermore, fog computing permits application designers to support analytics and instantaneous data that is actionable intuitions direct from IoT end terminal devices with least data exchange (on sites) and low latency, using client-based resources. Assume that the simplest gas usage meter counter as actual, direct feedback system can sufficiently impact client behaviour with regard to gas usage, resulting more modified and sustainable behaviour.

It's obvious that the personal usage of gas consumption data has crucial significance in smart metering applications. In present smart grid applications where meter communicates all measured data to cloud-based services, giving secondary importance to privacy requirement of data.

Due to this, naturally the personal data can be retrieved from well known measurements. Clients' lifestyle can be retrieved easily from detailed information acquired of gas utilization, revels information about home stay timings, meal and working schedules or even religious practices. Only solution to this is to provide strict data security [9] system for smart grids that go along with the fog computing protocols to secure the privacy of data collected.

Analysis indicates that slow end devices contributing to fog computing architecture can implement advanced cryptographic mechanism in an energy effective way. Adopting such mechanisms will support in securing the clients data privacy along with minimizing the communication and storage overheads [10]. The hierarchical structure of the fog computing architecture in addition with supportive Hyper elliptic curve identity based proxy signcryption scheme for Smart Gas Distribution Grid in fog computing environment (HYEC-IBPSC-SGDG-FC) scheme safeguards client's privacy from third partners. Then again differential privacy methods [11] can be utilized to implement effective secrecy preserving techniques for load management.

Modern consumption patterns have been built recently considering fog computing components of the architecture to merge noise to the data acquired at particular points, so supporting a best trade off between usefulness of the data and secrecy assessed with other popular techniques.

The significant contributions of this paper are as follows:

- A new secure Hyper elliptic curve identity based proxy signcryption scheme for Smart Gas Distribution Grid in fog computing environment (HYEC-IBPSC-SGDG-FC) scheme is proposed for smart gas grid network.
- Security analysis of proposed HYEC-IBPSC-SGDG-FC proves that the proposed scheme withstands HYEC-DLP and HYEC-DHP.
- The performance analysis of the proposed HYEC-IBPSC-SGDG-FC scheme is done by using the well-known AVISPA tool shows that the proposed scheme has resilience against replay and man-in-the-middle attacks.
- Finally the comparison with existing scheme shows that, HYEC-IBPSC-SGDG-FC is more efficient in terms of computation and communication costs.

The rest of the paper is organized as follows**.** Section 2 contains a discussion of the existing schemes that are related to this topic. In Sect. 3, the proposed scheme's characteristics and security assumptions are discussed. All phases of the proposed scheme are described in detail, and the proposed HYEC-IBPSC-SGDG-FC scheme is also discussed in detail. The meticulous security analysis and correctness of the proposed scheme is discussed in Sect. 4. Section 5 deals with the formal verification of HYEC-IBPSC-SGDG-FC using Automated Validation of Internet Security Protocols and Applications (AVISPA) tool and also discusses the assessment of performance. Finally, in Sect. 6, we wrap up our investigation.

2 Related work

In 1996, the authors of Ref. [12] proposed the conception of a proxy signature for the first time. The Original signer delegates the signing authority to proxy signer, and proxy signer issued a valid signature on behalf of original signer in accordance with that delegated authority The signcryption algorithm and the proxy signature concept come together to form proxy signcryption. Using an ID-based proxy encryption scheme was proposed by the authors of Ref. [13] in 2004. We found that this scheme did not meet the necessities of unforgeability and forward security. An improved IDPS system without a secure channel was developed by authors of Ref. [14] a year later, in 2005. An identity-based proxy signature was created by authors of Ref. [14] using bilinear pairing in the same year. In the proposed scheme, bilinear pairing was also used, which is a computationally demanding process. There is an ID-PSC (ID-PSC) scheme proposed by the authors of Ref. [15]. This is a public-verifiable, forward secure and computationally efficient scheme. By employing the universally composable (UC) paradigm, the authors of Ref. [16] developed an identity-based proxy sign encryption scheme (IBPSP). The authors of this scheme have provided a proof of semantic security of proposed scheme. To further protect the cloud delegation process, an identity-based signcryption mechanism is described in Ref. [17]. Encrypted messages are generated by the proxy agent and sent to the CSP, where they can be decrypted and checked. Due to the use of bilinear pairing in Ref. [18], the proposed solution was not suitable for drones [19]. Proposes a novel ECC-based IBPS approach to reduce the computational burden of the bilinear pairing approach. This was followed by the authors of Ref. [20] who proposed the use of IBPS scheme for drones, which they claimed to be simpler and more consistent than preceding approaches. a light weight and secure proxy blind signcryption for multi-digital messages based on a hyperelliptic curve (HEC) is proposed by authors of Ref. [21]. Our research essentially adds to that of scheme proposed in Ref. [20]. The use of HECC, which only needs a key size of 80 bits, is a big advantage of our scheme. ECC and bilinear pairing require a lot more keys [22].

3 Proposed HYEC-IBPSC-SGDG-FC scheme

The system model for the proposed HYEC-IBPSC-SGDG-FC is described in Fig. 2.

3.1 Preliminaries

The foundation and fundamental principles of hyper elliptic curve [23, 24], assumption of complexity, nomenclature, and the mathematical formulation of the proposed HYEC-IBPSC-SGDG-FC will be discussed in this segment. Table 1 lists the notations that were utilized in this work.

Definition 1 The hyperelliptic curve discrete logarithm problem (HYEC-DLP)

Given a $\mathfrak{H}y\mathcal{E}$ of genus G, the element $\mathcal{D}$ of order $\mathbb{N}$ of Jacobian, the other element $\mathcal{D}_1$ from the subgroup of $\mathcal{D}$. The HYEC-DLP is to extract the value of $\mathbb{N}$.

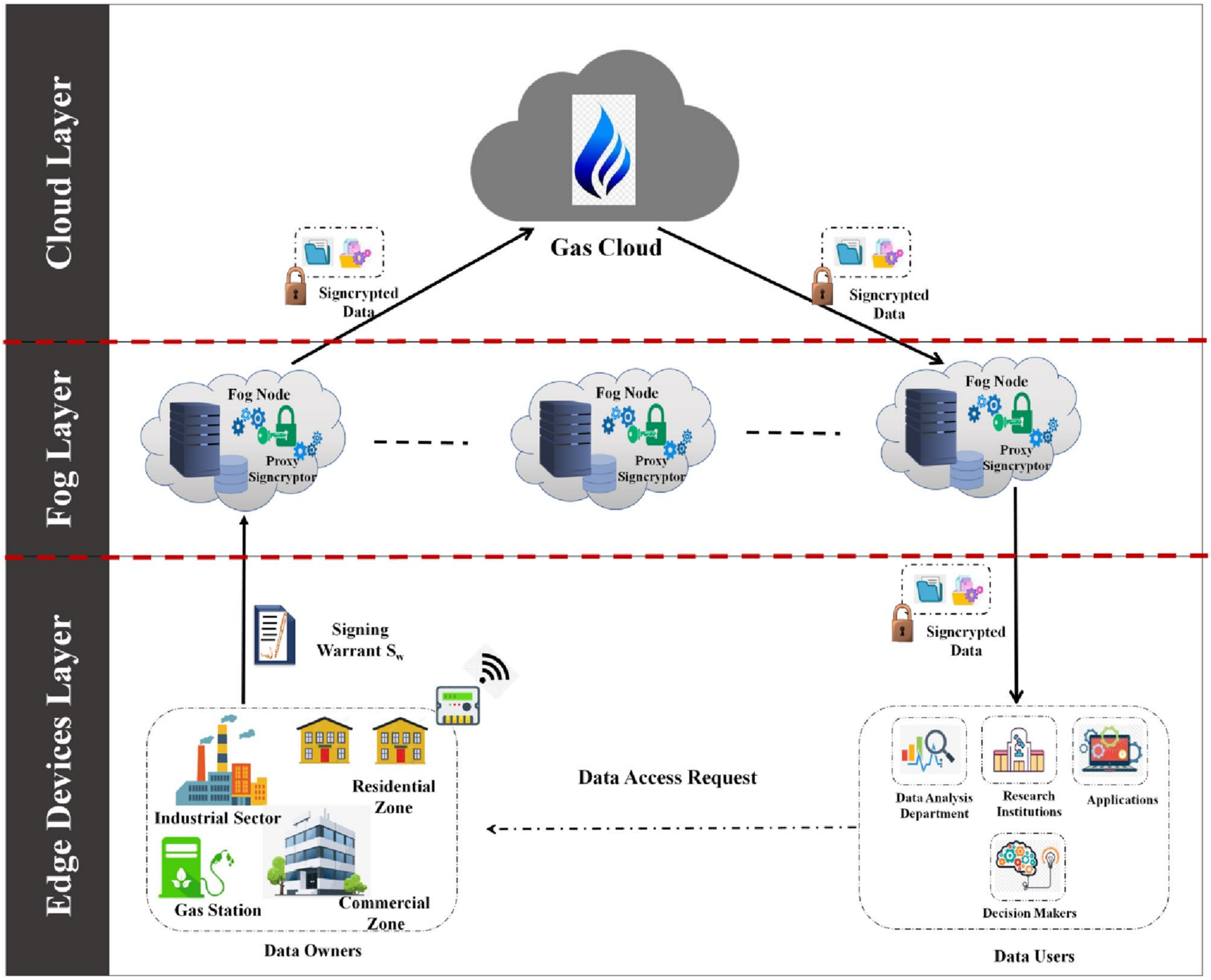


Fig. 2 The system model for the proposed HYEC-IBPSC-SGDG-FC scheme

Table 1 Notations used in this work

Notation	Meaning
$\mathfrak{H}y\mathcal{E}$	Hyper elliptic curve
$\boldsymbol{M_{Pub}}, \boldsymbol{\vartheta}$	Master Public and private key
$\mathscr{H}_{1,}\mathscr{H}_{2,}\mathscr{H}_{3,}\mathscr{H}_{4}$	Irreversible cryptographic hash function
$_{\mathrm{o}}$	data owner
$\mathrm{P_S}$	proxy signer
$_{\mathrm{R}}$	data requester
$\boldsymbol{S_{\mathcal{D}_o}}, \boldsymbol{P_{\mathcal{D}_o}}$	Secret key and public key of data owner
$\boldsymbol{S_{Ps}}, \boldsymbol{P_{Ps}}$	Secret key and public key of proxy signer
$\boldsymbol{S_{\mathcal{D}_R}}, \boldsymbol{P_{\mathcal{D}_R}}$	Secret key and public key of data requester
$\mathrm{S_w}$	Signcrypting warrant
w	Message warrant
$\boldsymbol{\Psi}$	Signcrypted message
$\mathcal{D}$	devisor of $\mathfrak{H}y\mathcal{E}$
$\oplus$	Bitwise XOR operation

Definition 2 The hyperelliptic curve Diffie-Hellman problem (HYEC-DHP)

Given a $\mathfrak{H}y\mathcal{E}$ of genus G, the element $\mathcal{D}$ of order $\mathbb{N}$ of Jacobian, the other elements $\mathbb{N} * \mathcal{D}_1$ and $\mathrm{P}^*\mathcal{D}_2$ from the subgroup of $\mathcal{D}$. The HYEC-DHP is to extract the value of $\mathbb{N}$ and P.

3.2 Formal model

The HYEC-IBPSC-SGDG-FC Scheme is divided into seven phases as follows. The sequence diagram in Fig. 3 describes the flow and phases of HYEC-IBPSC-SGDG-FC Scheme.

1. Phase 1: System Initialization—This algorithm is accountable for generating public parameters which are openly accessible to all the participating entities and master secret which is a secret of the trusted third party.
2. Phase 2: Key Extraction—Every individual user sends his/her unique identity $\mathrm{ID_i}$ to the trusted third party. The secret key for the user $_{\mathrm{o}}$ with identity $\mathrm{ID}_{\mathcal{D}_o}$ is $S_{\mathcal{D}_o}$ and public key is $P_{\mathcal{D}_o}$.
3. Phase 3: Warrant generation and Delegation -The original signer shall make a warrant w which contains the information about the type of delegation and time of delegation; it also defines the type of documents to be signcrypted by proxy signcryptor. This algorithm is accountable for generating the signing warrant $\mathrm{S_w}$ and delegating it to proxy signer.
4. Phase 4: Warrant Verification-This phase is accountable for the verification of signing warrant received from original signer. If the warrant is verified correctly then the proxy signer executes the next algorithm.
5. Phase 5: Proxy Signcryption-This phase takes the message to be sent M, proxy signers identity $\mathrm{ID_p}$, proxy Signers private key S_{Ps} identity of receiver $\mathrm{ID}_{\mathcal{D}_R}$ and public parameters as input and generates the signcrypted message and send to the receiver via a secure channel.
6. Phase 6: Unsigncryption—This algorithm takes received signcrypted message, receivers private key $S_{\mathcal{D}_R}$ and the identity of both sender and receiver $\mathrm{ID}_{\mathrm{P_s}}$, $\mathrm{ID}_{\mathcal{D}_R}$ and generates the original message M if the signcrypted message has not tampered else it returns ⊥ ..

3.3 Proposed scheme

3.3.1 Phase 1: system initialization

Input:-$\mathfrak{H}y\mathcal{E}$ Security parameters λ

Output:-public system parameters

1. Select $\vartheta \in_R \mathbb{Z}_n$, where ϑ is a master secret
2. Compute Master Public key $M_{Pub} = \vartheta * \mathcal{D}$, where $\mathcal{D}$ is devisor of $\mathfrak{H}y\mathcal{E}$
3. Select irreversible cryptographic hash functions $\mathscr{H}_{1,}\mathscr{H}_{2,}\mathscr{H}_{3,}\mathscr{H}_{4}$
4. The PKG publish the public system parameters as $= \{M_{Pub}, \mathcal{D}, \mathfrak{H}y\mathcal{E}, \mathscr{H}_{1,}\mathscr{H}_{2,}\mathscr{H}_{3,}\mathscr{H}_{4}, \mathrm{n} \geq 2^{80}\}$

3.3.2 Phase 2: key extraction

Input:-Identity of Participating entities $\mathrm{ID_i}$

Output:-Public and secret keys for $\mathrm{ID_i}$1. For the data owner $_{\mathrm{o}}$ with identity $\mathrm{ID}_{\mathcal{D}_o}$ PKG Selects $\varphi_{\mathcal{D}_o} \in_R \mathbb{Z}_n$

2. The Public key of data owner $_{\mathrm{o}}$ is $P_{\mathcal{D}_o} = \varphi_{\mathcal{D}_o} * \mathcal{D}$
3. Compute $\sigma_{\mathcal{D}_o} = \mathscr{H}_1\left(\mathrm{ID}_{\mathcal{D}_o}, P_{\mathcal{D}_o}\right)$
4. The Secret key of data owner $_{\mathrm{o}}$ with identity $\mathrm{ID}_{\mathcal{D}_o}$ is $S_{\mathcal{D}_o} = \varphi_{\mathcal{D}_o} + \sigma_{\mathcal{D}_o} * \vartheta$
5. The Secret key of proxy signer $\mathrm{P_S}$ with identity $\mathrm{ID}_{\mathrm{P_s}}$ is $S_{\mathrm{Ps}} = \varphi_{\mathrm{Ps}} + \sigma_{\mathrm{Ps}} * \vartheta$
6. The Secret key of data requester $_{\mathrm{R}}$ with identity $\mathrm{ID}_{\mathcal{D}_R}$ is $S_{\mathcal{D}_R} = \varphi_{\mathcal{D}_R} + \sigma_{\mathcal{D}_R} * \vartheta$

3.3.3 Phase 3: warrant generation and delegation

Input:-public system parameters, $S_{\mathcal{D}_o}$,w

Output:-Signcrypting warrant $\mathrm{S_w}$

The original signer shall make a warrant w which contains the information about the type of delegation and time of delegation; it also defines the type of documents to be signcrypted by proxy signcryptor.

By using warrant w the original signer generates signcrypting warrant $\mathrm{S_w}$ by using original signer's private key $S_{\mathcal{D}_o}$

1. Select $\alpha \in_R \mathbb{Z}_n$

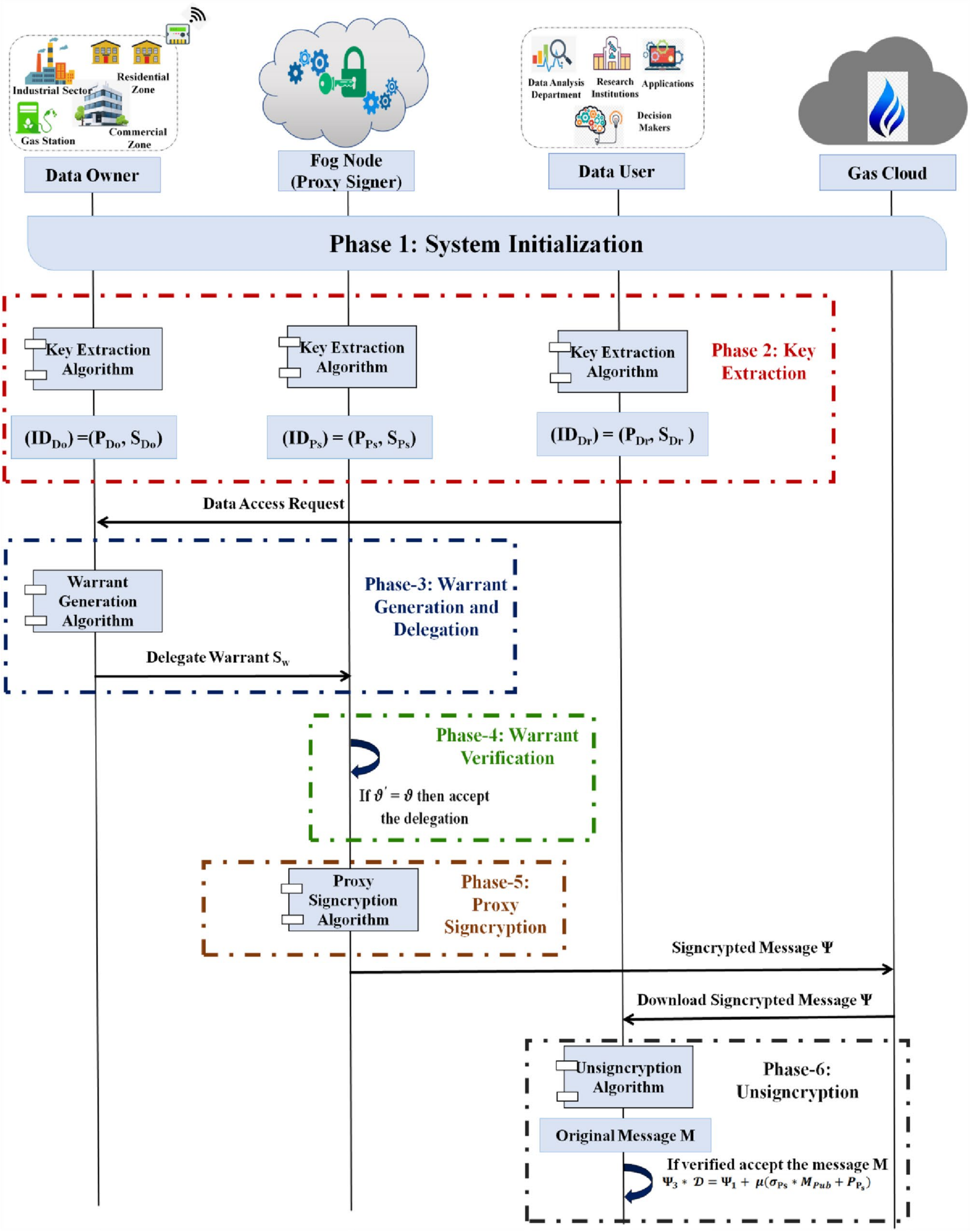

Industrial Sector
Residential Zone
Gas Station
Commercial Zone
Data Owner
Fog Node (Proxy Signer)
Data Analysis Department
Research Institutions
Applications
Decision Makers
Data User
Gas Cloud
Phase 1: System Initialization
Key Extraction Algorithm
Key Extraction Algorithm
Key Extraction Algorithm
Phase 2: Key Extraction
$(ID_{Do}) = (P_{Do}, S_{Do})$
$(ID_{Ps}) = (P_{Ps}, S_{Ps})$
$(ID_{Dr}) = (P_{Dr}, S_{Dr})$
Data Access Request
Warrant Generation Algorithm
Phase-3: Warrant Generation and Delegation
Delegate Warrant S_w
Phase-4: Warrant Verification
If $\vartheta' = \vartheta$ then accept the delegation
Proxy Signcryption Algorithm
Phase-5: Proxy Signcryption
Signcrypted Message Ψ
Download Signcrypted Message Ψ
Unsigncryption Algorithm
Phase-6: Unsigncryption
Original Message M
If verified accept the message M
$\Psi_3 * \mathcal{D} = \Psi_1 + \mu(\sigma_{Ps} * M_{Pub} + P_{P_s})$

Fig. 3 Sequence diagram representing flow and phases of HYEC-IBPSC-SGDG-FC Scheme

2. Compute $\mathrm{v} = \alpha * \mathcal{D}$
3. Compute$\partial = \mathscr{H}_2\,(\mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, \mathrm{w}, \mathrm{v})$
4. Compute $\mathrm{S_w} = \alpha + \partial * S_{\mathcal{D}_o}$
5. The original signer sends $\mathrm{W} = (\mathrm{S_w}, \mathrm{v}, \mathrm{w})$ to proxy signcryptor

3.3.4 Phase 4: warrant verification

Input:-$\mathrm{W} = (\mathrm{S_w}, \mathrm{v}, \mathrm{w})$.

Output:-Accept or reject the signing warrant.

1. Compute $\delta^{\}} = \mathscr{H}_2\,(\mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, \mathrm{w}, \mathrm{v})$
2. The proxy signer verifies the received delegation by computing

$$\mathrm{S_w} * \mathcal{D}\ominus = \nu + \partial * \left(\sigma_{\mathcal{D}_o} * M_{Pub} + P_{\mathcal{D}_o}\right) \tag{1}$$

Verification of Eq (1)
Consider LHS, Substitute $\mathrm{S_w} = \alpha + \partial * S_{\mathcal{D}_o}$
$\mathrm{S_w} * \mathcal{D} = \mathcal{D} * \left(\alpha + \partial * S_{\mathcal{D}_o}\right)$
$\mathrm{S_w} * \mathcal{D} = \mathcal{D} * \alpha + \mathcal{D} * \partial * S_{\mathcal{D}_o}$
Substitute $\mathrm{v} = \alpha * \mathcal{D}$ and $S_{\mathcal{D}_o} = \varphi_{\mathcal{D}_o} + \sigma_{\mathcal{D}_o} * \vartheta$
$\mathrm{Sw} * \mathcal{D} = \mathrm{v} + \mathcal{D} * \partial * (\varphi_{\mathcal{D}_o} + \sigma_{\mathcal{D}_o} * \vartheta)$
$\mathrm{Sw} * \mathcal{D} = \mathrm{v} + \partial * (\mathcal{D} * \varphi_{\mathcal{D}_o} + \mathcal{D} * \sigma_{\mathcal{D}_o} * \vartheta)$
Substitute $M_{Pub} = \vartheta * \mathcal{D}$ and $P_{\mathcal{D}_o} = \varphi_{\mathcal{D}_o} * \mathcal{D}$
$\mathrm{Sw} * \mathcal{D} = \mathrm{v} + \partial * (\sigma_{\mathcal{D}_o} * M_{Pub} + P_{\mathcal{D}_o})$
Hence proved Eq. 1

3.3.5 Phase 5: proxy signcryption

If warrant is verified in previous step, the proxy signcryptor then signcrypts the message.

Input:-Public system parameters, m,W, $\sigma_{\mathcal{D}_R}$,$P_{\mathcal{D}_R}$

Output:-Signcrypted message Ψ1. Select $\rho \in_R \mathbb{Z}_{p^*}$
2. $\Psi_1 = \rho * \mathcal{D}$
3. *Compute*$\mathcal{Q} = \rho * (\sigma_{\mathcal{D}_R} * M_{Pub} + P_{\mathcal{D}_R})$
4. *Compute* ƙ $= \mathscr{H}_3\,(\Psi_1, \mathcal{Q}, \mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{ID}_{\mathcal{D}_R}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, P_{\mathcal{D}_R})$
5. $\Psi_2 = \mathrm{m} \oplus$ ƙ
6. *Compute*$\mu = \mathscr{H}_4\,(\mathrm{m}, \mathrm{W}, \mathcal{Q}, \Psi_1, \mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{ID}_{\mathcal{D}_R}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, P_{\mathcal{D}_R})$
7. $\Psi_3 = \rho + \mu * S_{\mathrm{Ps}}$
8. $\Psi = \left(\Psi_1, \Psi_2, \Psi_3, W\right)$

The proxy signcryptor uploads the signcrypted ciphertext Ψ on cloud.

3.3.6 Phase 6: unsigncryption

Input:-Public system parameters, $S_{\mathcal{D}_R}$, Ψ

Output:-Original message m or ⊥

The receiver with identity $\mathrm{ID}_{\mathcal{D}_R}$ will download the signcrypted ciphertext Ψ from cloud and perform the following operations to compute the original message m.

1. Compute $\mathrm{v} = S_{\mathcal{D}_R} * \Psi_1$
2. Compute ƙ $= \mathscr{H}_3\,(\mathrm{v}, \Psi_1, \mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{ID}_{\mathcal{D}_R}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, P_{\mathcal{D}_R})$
3. Compute $\mathrm{m} = \Psi_2 \oplus$ ƙ
4. Compute $\mu = \mathscr{H}_4\,(\mathrm{m}, \mathrm{W}, \mathrm{v}, \Psi_1, \mathrm{ID}_{\mathcal{D}_o}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{ID}_{\mathcal{D}_R}, P_{\mathcal{D}_o}, P_{\mathrm{P_s}}, P_{\mathcal{D}_R})$
5. Verify whether

$$\Psi_3 * \mathcal{D} = \Psi_1 + \mu\left(\sigma_{\mathrm{Ps}} * M_{Pub} + P_{\mathrm{P_s}}\right) \tag{2}$$

Verification of Eq. (2)
Consider LHS, substitute $\Psi_3 = \rho + \mu * S_{\mathrm{Ps}}$
$\Psi_3 * \mathcal{D} = (\rho + \mu * S_{\mathrm{Ps}}) * \mathcal{D}$
$\Psi_3 * \mathcal{D} = (\rho * \mathcal{D} + \mu * S_{\mathrm{Ps}} * \mathcal{D})$
Substitute $\Psi_1 = \rho * \mathcal{D}$ and $S_{\mathrm{Ps}} = \varphi_{\mathrm{Ps}} + \sigma_{\mathrm{Ps}} * \vartheta$
$\Psi_3 * \mathcal{D} = \left(\Psi_1 + \mu * \left(\varphi_{\mathrm{Ps}} + \sigma_{\mathrm{Ps}} * \vartheta\right) * \mathcal{D}\right)$
$\Psi_3 * \mathcal{D} = \left(\Psi_1 + \mu * \left(\varphi_{\mathrm{Ps}} * \mathcal{D} + \sigma_{\mathrm{Ps}} * \vartheta * \mathcal{D}\right)\right)$
Substitute$P_{\mathrm{P_s}} = \varphi_{P_s} * \mathcal{D}$ and $M_{Pub} = \vartheta * \mathcal{D}$

$$\Psi_3 * \mathcal{D} = \Psi_1 + \mu(\sigma_{\mathrm{Ps}} * M_{Pub} + P_{\mathrm{P_s}})$$

Hence proved.

4 Security model

The proposed HYEC-IBPSC-SGDG-FC scheme should assure confidentiality and unforgeability of original message. Let us consider that there exist an adversary $\mathcal{A}_d$ for the proposed scheme and $\mathbb{C}_\mathrm{h}$ is a challenger.

Game-1

The following game is played between adversary $\mathcal{A}_d$ and challenger $\mathbb{C}_\mathrm{h}$ to solve the problem of HYEC-DHP.

Initialization

The challenger $\mathbb{C}_\mathrm{h}$ runs the setup phase to generate the public parameters and a master secret ϑ.Then $\mathbb{C}_\mathrm{h}$ forward the public parameters to adversary $\mathcal{A}_d$ and keeps ϑ with itself.

Phase 1: Adversary $\mathcal{A}_d$ executes the following queries which are interdependent.

1. Hash Function query:- Adversary $\mathcal{A}_d$ can request for any hash function value.
2. Key Extraction query:-Adversary $\mathcal{A}_d$ selects the unique identity as ID and requests for public and secret key. The challenger $\mathbb{C}_\mathrm{h}$ runs key extraction algorithm and returns the public and secret key to Adversary $\mathcal{A}_d$.
3. Warrant generation and Delegation query:-The adversary $\mathcal{A}_d$ sends the request for signing warrant. The challenger $\mathbb{C}_\mathrm{h}$ returns the warrant w and signing warrant $\mathrm{S_w}$
4. Warrant Verification query:-The adversary $\mathcal{A}_d$ verifies the signing warrant received from challenger $\mathbb{C}_\mathrm{h}$

5. Proxy Signcryption query:-The adversary $\mathcal{A}_d$ selects message m and the identities $\mathrm{ID}_{\mathcal{D}_o}$, $\mathrm{ID}_{\mathrm{P_s}}$ and $\mathrm{ID}_{\mathcal{D}_R}$. The challenger $\mathbb{C}_h$ executes Proxy Signcryption and sends the signcrypted ciphertext Ψ to$\mathcal{A}_d$.
7. Unsigncryption query:-The adversary $\mathcal{A}_d$ selects the signcrypted ciphertext Ψ and the identities $\mathrm{ID}_{\mathcal{D}_o}$, $\mathrm{ID}_{\mathrm{P_s}}$ and$\mathrm{ID}_{\mathcal{D}_R}$. The challenger $\mathbb{C}_h$ then executes Unsigncryption algorithm and sends result to$\mathcal{A}_d$.

Challenge: The adversary $\mathcal{A}_d$ wishes to be challenged on the two messages M_0, M_1 and identities $\mathrm{ID}_{\mathrm{P_s}}$ and$\mathrm{ID}_{\mathcal{D}_R}$. The challenger $\mathbb{C}_h$ produces the random bit Ь $\in_R$ {0,1} for which the $\Psi = (\Psi_1, \Psi_2, \Psi_3, W)$ and sends to$\mathcal{A}_d$. The adversary $\mathcal{A}_d$ executes the queries like $\mathscr{H}$ queries, Key Extraction query, Warrant generation and Delegation query Proxy Signcryption query and Unsigncryption query.

Guess:-The adversary $\mathcal{A}_d$ produces he random bit Ь$'$$\in_R$ {0,1}. If Ь = Ь$'$ the adversary $\mathcal{A}_d$ wins the game. We have following advantage of $\mathcal{A}_d$

$$\mathrm{Adv}\ (\mathcal{A}_d) = \ Pr[Ь = Ь'] - \frac{1}{2}$$

Game-2

The following game is played between adversary $\mathcal{A}_d$ and challenger $\mathbb{C}_h$ to solve the problem of HYEC-DLP.

Setup

The challenger $\mathbb{C}_h$ executes the setup algorithm in order to obtain the public parameters and a master secret ϑ. Then $\mathbb{C}_h$ sends adversary $\mathcal{A}_d$ the public parameters .

Queries

Then $\mathcal{A}_d$ performs polynomial limited number of queries like in HYEC-DHP.

Forgery

Finally, adversary $\mathcal{A}_d$ generates (Ψ,$\mathrm{ID}_{\mathcal{D}_o}$,$\mathrm{ID}_{\mathrm{P_s}}$), In phase 2 the private key for $\mathrm{ID}_{\mathcal{D}_o}$ was not asked and the adversary $\mathcal{A}_d$ wins the game if the output of Unsigncryption ($\Psi, S_{\mathcal{D}_o}$,$\mathrm{ID}_{\mathrm{P_s}}$) is not$\perp$.

4.1 Security analysis

In this section the proof of above two games is described. The subsequent Games describes that when the game is played between adversary $\mathcal{A}_d$ and challenger $\mathbb{C}_h$ how it provides confidentiality and unforgeability.

Game-1

If the adversary $\mathcal{A}_d$ possesses the ability to create two genuine jumbled texts in this game and having acceptable advantage $\mathrm{Adv}(\mathcal{A}_d)$ and execute maximum $\mathcal{Q}_{\mathcal{H}\mathrm{i}}$ queries, $\mathcal{Q}_{ke}$ key extraction queries includes ($\mathcal{Q}_{pk}$, $\mathcal{Q}_{sk}$) Public key and secret key queries respectively. Warrant generation and Delegation query $\mathcal{Q}_{gd}$ and proxy signcryption queries $\mathcal{Q}_{psc}$. Then challenger $\mathbb{C}_h$ can solve HYEC-DHP with the advantage of

$$\mathrm{Adv}(\mathcal{A}_d)^* \geq \mathrm{Adv}(\mathcal{A}_d)\left(1 - \frac{\mathcal{Q}_{sk}}{\mathcal{Q}_{pk}}\right)\left(1 - \frac{1}{2^{\lambda}}\right)\left(\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}\right)$$

Proof If the challenger $\mathbb{C}_h$ selects the two random numbers $_{1,\ 2}$ then the $\mathbb{C}_h$ has to solve the $_1{*}\mathcal{D} = {}_2{*}\mathcal{D} = {}_1{*}\ {}_2{*}\mathcal{D}$ for adversary $\mathcal{A}_d$. □

Setup: the challenger $\mathbb{C}_h$ sets $= \{M_{Pub}, \mathcal{D}$, $\mathfrak{H}$yƐ, $\mathscr{H}_{1,}$ $\mathscr{H}_{2,}$ $\mathscr{H}_{3,}$ $\mathscr{H}_4$, n$\geq 2^{80}\}$ as a public system parameters and sends to adversary $\mathcal{A}_d$.

Queries: The adversary $\mathcal{A}_d$ asks for the subsequent queries.

$\mathbf{H_1}$ queries: The adversary $\mathcal{A}_d$ asks for the $(\mathrm{ID_i}, P_\mathrm{i}, \sigma_\mathrm{i})$, the challenger $\mathbb{C}_h$ reply with σ_i if it exists in the list ($\mathscr{L}\mathscr{H}_1$), or else reply with a randomly selected value and add $(\mathrm{ID}_{\mathcal{D}_o}, P_{\mathcal{D}_o}, \sigma_{\mathcal{D}_o})$ to $\mathscr{L}\mathscr{H}_{1.}$

$\mathbf{H_2}$ queries: The adversary $\mathcal{A}_d$ asks for the ($\mathrm{ID_i}, P_\mathrm{i}$, w, v, ∂), the challenger $\mathbb{C}_h$ reply with ∂ if it exists in the list ($\mathscr{L}\mathscr{H}_2$), or else reply with a randomly selected value and add ($\mathrm{ID_i}, P_\mathrm{i}$, w, v, ∂) to $\mathscr{L}\mathscr{H}_{2.}$

$\mathbf{H_3}$ queries: The adversary $\mathcal{A}_d$ asks for the (Ψ_1, $\mathcal{Q}$, $\mathrm{ID_i}, P_\mathrm{i}$, ƙ), the challenger $\mathbb{C}_h$ reply with ƙ if it exists in the list ($\mathscr{L}\mathscr{H}_3$), or else reply with a randomly selected value and add (Ψ_1, $\mathcal{Q}$, $\mathrm{ID_i}, P_\mathrm{i}$, ƙ) to $\mathscr{L}\mathscr{H}_{3.}$

$\mathbf{H_4}$ queries: The adversary $\mathcal{A}_d$ asks for the (m, W, $\mathcal{Q}$, Ψ_1, $\mathrm{ID_i}, P_\mathrm{i}, \mu$), the challenger $\mathbb{C}_h$ reply with μ if it exists in the list ($\mathscr{L}\mathscr{H}_4$), or else reply with a randomly selected value and add (m, W, $\mathcal{Q}$, Ψ_1, $\mathrm{ID_i}, P_\mathrm{i}, \mu$) to $\mathscr{L}\mathscr{H}_{4.}$

Key extraction queries: key extraction queries includes ($\mathcal{Q}_{pk}$,$\mathcal{Q}_{sk}$) Public key and secret key queries respectively. When adversary $\mathcal{A}_d$ asks for $\mathcal{Q}_{pk}$ if $\mathrm{ID_i}$ =$\mathrm{ID_j}$, the challenger $\mathbb{C}_h$ sets P_i=\\$_1 * \mathcal{D}$, or else it will execute P_i=$\varphi_i * \mathcal{D}$, where$\varphi_i \in \{1, 2, 3 \ldots \ldots .n\}$. Then update$\mathcal{L}_{pk}$. When adversary $\mathcal{A}_d$ asks for $\mathcal{Q}_{sk}$, if $\mathrm{ID_i}$ =ID_*, the challenger $\mathbb{C}_h$ terminates the execution, or else sets $S_{\mathrm{i}=}\ \varphi_i + \sigma_\mathrm{i} * \vartheta$ and reply the adversary$\mathcal{A}_d$. Then update$\mathcal{L}_{sk}$.

Warrant generation and delegation query: The adversary $\mathcal{A}_d$ asks for the $\mathcal{Q}_{gd}$, if $\mathrm{ID}_{\mathcal{D}_o}$=$\mathrm{ID}_*$, the the challenger $\mathbb{C}_h$ reply with W using Warrant generation and Delegation algorithm to the adversary $\mathcal{A}_d$, or else it calculates v = w+$\delta\left(\sigma_{\mathcal{D}_o} * M_{Pub} + P_{\mathcal{D}_o}\right)$ where δ, w $\in \{1, 2, 3 \ldots \ldots .n\}$, then set W = ($\mathrm{S_w}$, v, w) and reply to adversary $\mathcal{A}_d$.

Proxy signcryption query: If the adversary $\mathcal{A}_d$ asks and provides Message M with $\mathrm{ID}_{\mathcal{D}_o}$,$\mathrm{ID}_{\mathrm{P_s}}$ and $\mathrm{ID}_{\mathcal{D}_R}$, if $\mathrm{ID}_{\mathrm{P_s}}$ =ID_*, then the challenger $\mathbb{C}_h$ reply as it computes $\Psi_1 = \rho * \mathcal{D}$ and $\mathcal{Q} = \rho * (\sigma_{\mathcal{D}_R} * M_{Pub} + P_{\mathcal{D}_R})$ where $\rho \in Z_n$, compute Ψ_2= m$\oplus$ ƙ, where ƙ $\in Z_n$, compute Ψ_3= $\rho + \mu * S_{\mathrm{Ps}}$, where $\mu \in Z_n$, and forward $\Psi^* = (\Psi_1, \Psi_2, \Psi_3, W)$ to adversary $\mathcal{A}_d$. Or else it replies by calling signcryption algorithm.

Unsigncryption query: If the adversary $\mathcal{A}_d$ asked, if $\mathrm{ID}_{\mathrm{D_R}} \neq \mathrm{ID}_*$, then the challenger $\mathbb{C}_\mathrm{h}$ replied by calling unsigncryption algorithm.

Challenge: An adversary $\mathcal{A}_d$ may outputs two messages M_0 and M_1, and two identities $\mathrm{ID}_{\mathrm{P_s}}$ and $\mathrm{ID}_{\mathcal{D}_\mathrm{R}}$, if $\mathrm{ID}_{\mathrm{P_s}} = \mathrm{ID}_*$, the challenger $\mathbb{C}_\mathrm{h}$ selects b ϵ 0,1 responds as, it calculates $\Psi_1 = \rho * \mathcal{D}$ and $\mathcal{Q} = \rho * (\sigma_{\mathcal{D}_\mathrm{R}} * M_{Pub} + P_{\mathcal{D}_\mathrm{R}})$ where $\rho \in \{1, 2, 3 \ldots \ldots .n\}$, compute $\Psi_2 = \mathrm{m} \oplus \Bbbk$, where $\Bbbk \in \{1, 2, 3 \ldots \ldots .n\}$, compute $\Psi_3 = \rho + \mu * S_{\mathrm{Ps}}$, where $\mu \in \{1, 2, 3 \ldots \ldots .n\}$, and send $\Psi^* = (\Psi_1, \Psi_2, \Psi_3, W)$ to adversary $\mathcal{A}_d$. Then the adversary $\mathcal{A}_d$ continue with $\mathscr{H}$ queries, Key Extraction query $(\mathcal{Q}_{ke})$, Warrant generation and Delegation query $(\mathcal{Q}_{gd})$, proxy signcryption queries $(\mathcal{Q}_{psc})$ and Un-signcryption query $(\mathcal{Q}_{usc})$.

Guess: An adversary $\mathcal{A}_d$ may output Ƅ′ = Ƅ, then adversary $\mathcal{A}_d$ is successful and identify the solution for HYEC-DHP instance, or else an adversary $\mathcal{A}_d$ terminate.

Then challenger $\mathbb{C}_\mathrm{h}$ can solve HYEC-DHP and be successful in challange phase and its probability as $\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}$ so we have the probability as.

$\mathrm{Adv}(\mathcal{A}_d)^* \geq \mathrm{Adv}(\mathcal{A}_d)\left(1 - \frac{\mathcal{Q}_{sk}}{\mathcal{Q}_{pk}}\right)\left(1 - \frac{1}{2^\lambda}\right)\left(\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}\right)$.

Game 2: The proposed HYEC-IBPSC-SGDG-FC scheme is unforgeable. If an adversary $\mathcal{A}_d$ has the capability of existential forgery for (EUF- HYEC-IBPSC-SGDG-FC- SPA) selected plaintext attack with acceptable advantage of $\mathrm{Adv}(\mathcal{A}_d)$. Then the challenger $\mathbb{C}_\mathrm{h}$ can solve HYEC-CDH with the advantage of $\mathrm{Adv}(\mathcal{A}_d)^* \geq \mathrm{Adv}(\mathcal{A}_d)\left(1 - \frac{\mathcal{Q}_{sk}}{\mathcal{Q}_{pk}}\right)\left(1 - \frac{1}{2^\lambda}\right)\left(\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}\right)$.

Proof If the challenger $\mathbb{C}_\mathrm{h}$ gets an instance of HYEC-CDH $(\mathcal{D}, \mathcal{D}.S_{\mathcal{D}_\mathrm{o}}, \mathcal{D}.S_{\mathrm{Ps}})$, then the challenger $\mathbb{C}_\mathrm{h}$ has to extract the values of$S_{\mathcal{D}_\mathrm{o}} and S_{\mathrm{Ps}}$. □

Setup: the challenger $\mathbb{C}_\mathrm{h}$ sets $= \{M_{Pub}, \mathcal{D}$, $\mathfrak{H}$y$\mathfrak{E}$, $\mathscr{H}_1, \mathscr{H}_2, \mathscr{H}_3, \mathscr{H}_4$, n$\geq 2^{80}$} as a public system parameters and sends to adversary $\mathcal{A}_d$.

Queries: The adversary $\mathcal{A}_d$ asks for $\mathcal{Q}_{\mathcal{H}i}$ queries, $\mathcal{Q}_{ke}$ key extraction queries includes $(\mathcal{Q}_{pk}, \mathcal{Q}_{sk})$ Public key and secret key queries respectively. Warrant generation and Delegation query $\mathcal{Q}_{gd}$ and $\mathcal{Q}_{psc}$ similar as Game 1.

Forgery: The adversary $\mathcal{A}_d$ generates the tuple $\{\mathrm{ID}_{\mathcal{D}_\mathrm{o}}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{W}\}$ or $\{\mathrm{W}, \mathrm{M_w}, \mathrm{ID}_{\mathcal{D}_\mathrm{o}}, \mathrm{ID}_{\mathrm{P_s}}, \mathrm{ID}_{\mathcal{D}_\mathrm{R}}\}$. The adversary $\mathcal{A}_d$ wins the game if the following cases hold.

Case-1: The challenger $\mathbb{C}_\mathrm{h}$ gets two delegation signatures $\mathrm{S_w} = \alpha + \partial * S_{\mathcal{D}_\mathrm{o}}$ and $S^*_w = \alpha + \partial^* * S_{\mathcal{D}_\mathrm{o}}$, so we have.

$\mathrm{S_w} - \alpha - \partial * S_{\mathcal{D}_\mathrm{o}} - (S^*_w - \alpha - \partial^* * S_{\mathcal{D}_\mathrm{o}}) = \mathrm{S_w} - \alpha - \partial * S_{\mathcal{D}_\mathrm{o}}$ - $S^*_w + \alpha + \partial^* * S_{\mathcal{D}_\mathrm{o}} = \mathrm{S_w} + S^*_w = \partial^* * S_{\mathcal{D}_\mathrm{o}} - \partial * S_{\mathcal{D}_\mathrm{o}} = \mathrm{S_w} + S^*_w = (\partial^* - \partial) * S_{\mathcal{D}_\mathrm{o}}$, So the private key can be extracted as $S_{\mathcal{D}_\mathrm{o}} = \frac{\mathrm{S_w} + S^*_w}{(\partial^* - \partial)}$

Case-2: The challenger $\mathbb{C}_\mathrm{h}$ gets two delegation signatures $\Psi_3 = \rho + \mu * S_{\mathrm{Ps}}$ and $\Psi_3{}^* = \rho + \mu^* * S_{\mathrm{Ps}}$, so we have.

$\Psi_3 - \rho - \mu * S_{\mathrm{Ps}} - (\Psi_3{}^* - \rho - \mu^* * S_{\mathrm{Ps}}) = \Psi_3 - \rho - \mu * S_{\mathrm{Ps}} - \Psi_3{}^* + \rho + \mu^* * S_{\mathrm{Ps}} = \Psi_3 + \Psi_3{}^* = \mu^* * S_{\mathrm{Ps}} - \mu * S_{\mathrm{Ps}} = (\mu^* - \mu)S_{\mathrm{Ps}}$. So the private key can be extracted as $S_{\mathrm{Ps}} = \frac{\Psi_3 + \Psi_3{}^*}{\mu^* - \mu}$

From the process, we can define 3 events as

E_1: The challenger $\mathbb{C}_\mathrm{h}$ is successful in executing queries with the probability of $\left(1 - \frac{\mathcal{Q}_{sk}}{\mathcal{Q}_{pk}}\right)$

E_2: The challenger $\mathbb{C}_\mathrm{h}$ is successful in proxy signcryption queries $\mathcal{Q}_{psc}$ with the probability of $\left(1 - \frac{1}{2^\lambda}\right)$

E_3: The $\mathrm{ID}_{\mathrm{P_s}} = ID^*$ with the probability of $\left(\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}\right)$

So the collective probability is

$\mathrm{Adv}(\mathcal{A}_d)^* \geq \mathrm{Adv}(\mathcal{A}_d)\left(1 - \frac{\mathcal{Q}_{sk}}{\mathcal{Q}_{pk}}\right)\left(1 - \frac{1}{2^\lambda}\right)\left(\frac{1}{\mathcal{Q}_{pk} - \mathcal{Q}_{sk}}\right)$

5 Performance analysis

In this section, the performance analysis of the proposed HYEC-IBPSC-SGDG-FC scheme is discussed. We use the well-known AVISPA tool [25, 26] to discuss the security proof and demonstrate that the proposed scheme is not susceptible to replay and man-in-the-middle attack. It should be noted that for any security protocol, AVISPA only handles replay and man-in-the-middle threats against an attacker.

The HLPSL [18] code is written for the proposed scheme with the different roles like original signer, proxy signer and trusted third party. This code is then executed using SPAN and AVISPA with the backends OFMC and CL-AtSe. We can see that no attacks were discovered by OFMC. In other words, for a limited number of sessions as specified in the role of the environment, the stated security goals were achieved. The proposed protocol is also executed with CL-AtSe backend for bounded number of sessions. The output shows that the protocol is safe under CL-AtSe also. The software resources such as Oracle VM Virtual Box and Security protocol animator (SPAN) are used. The output of AVISPA is shown in Figs. 4 and 5.

We have done the comparison of our proposed HYEC-IBPSC-SGDG-FC scheme with the existing proxy signcryption schemes [17, 19, 27–30]. The comparison outcomes are listed in Table 2. We define some notations as follows:

BPM:-Bilinear Pairing multiplications.

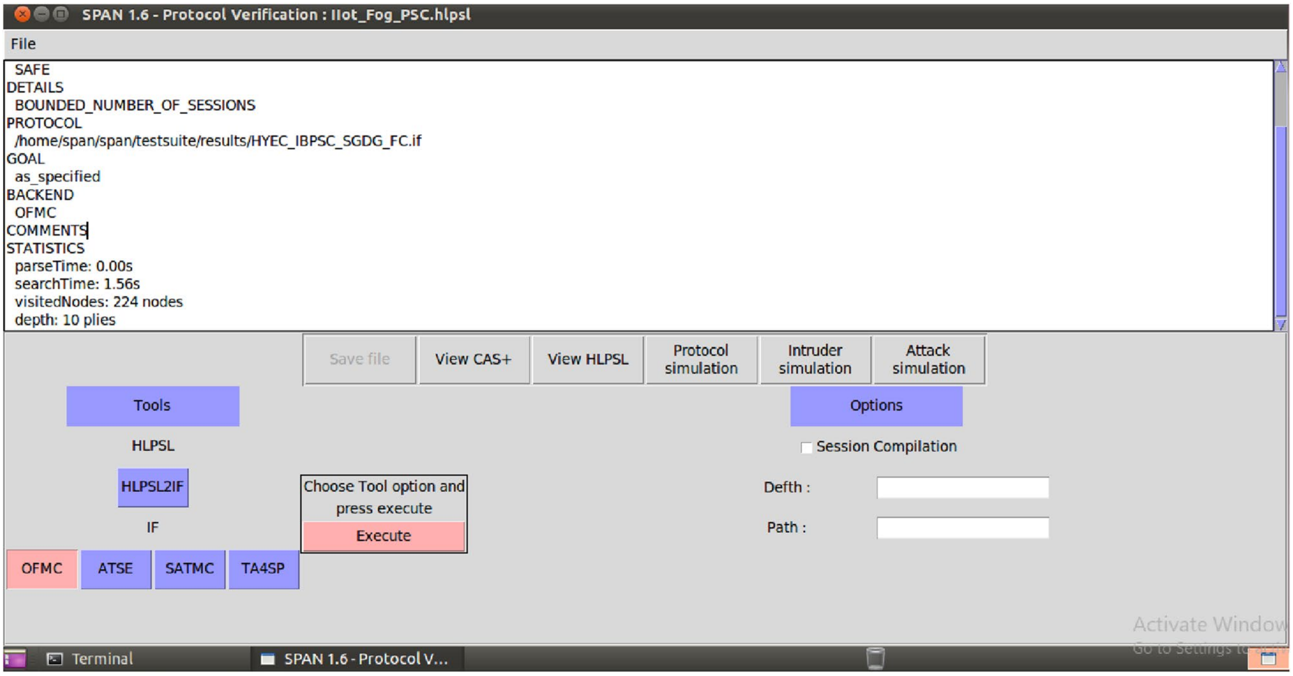
P:-Bilinear Pairing operation.

E:-exponentiation operation.

EPM:- elliptic curve point multiplication.

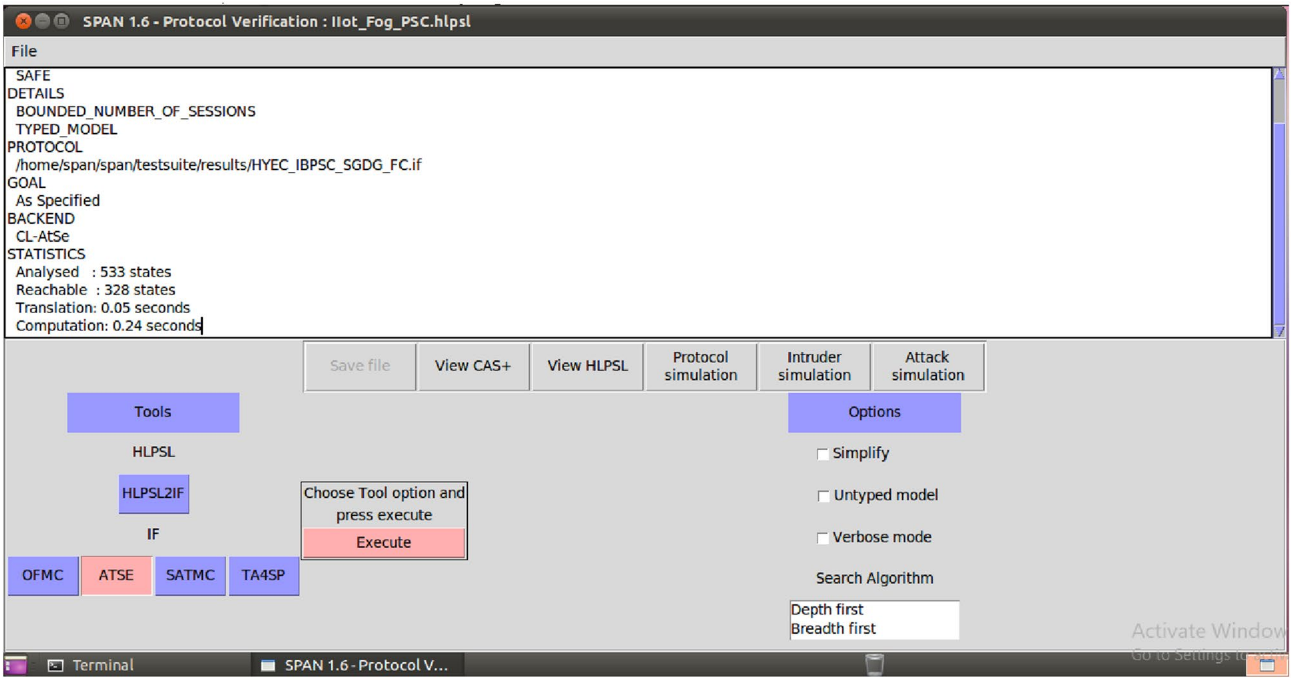
HDM:- hyperelliptic curve divisor multiplication.

The time required to perform the cryptographic operations are 14.90 ms for pairing operation, 4.31 ms for multiplication operation,1.25 ms for each exponentiation operation, 0.97 ms for elliptic curve point multiplication and 0.48 ms for hyperelliptic curve divisor multiplication.



```
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/HYEC_IBPSC_SGDG_FC.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 1.56s
 visitedNodes: 224 nodes
 depth: 10 plies
```

Fig. 4 OFMC output



```
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/HYEC_IBPSC_SGDG_FC.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed   : 533 states
 Reachable  : 328 states
 Translation: 0.05 seconds
 Computation: 0.24 seconds
```

Fig. 5 CL-AtSe output

To assess the computing efficiency of the various systems, we employ a simple technique. For example the scheme proposed by Ming [26] requires 11, 1Ɛ and 7 operations. Therefore the total time required for this scheme is 213.41 ms. In similar way the operation time required for each scheme is calculated and listed in Table 2.Hence it can be observed from Table 3, that HYEC-IBPSC-SGDG-FC significantly outperformed the alternative schemes describe in Refs. [18, 19, 26–29]. The comparison of computational costs in terms of time in milliseconds (ms) for each phase of the IDPSC schemes is shown graphically in Fig. 6

The comparison of communication cost is described in Tables 4 and 5. To calculate the communication cost we have considered that a single hash value ($\mathscr{H}$) is communicated it takes 512 bits, a message () is considered to be of 2048 bits and a pairing operation $(\cdot)$ is considered to be of 1024 bits, is considered to be of 160bits and N is considered to be of 80bits.Fig. 7 shows the details of communication cost comparison of various schemes with proposed scheme in bits.

Table 2 Computational cost comparison

Scheme	Warrant generation and delegation	Proxy secret key generation/ warrant verification	Proxy signcryption	Unsigncryption	Total
Ming and Wang [27]	3	2	3 + 4 + 3Ɛ	4M + 6	6 + 7Ɛ + 12
Zhou et al. [28]	3	2P + 1	3 + 1 + 1Ɛ	4 + 4	11 + 1Ɛ + 7
Yu and Wang [29]	1	1	1 + 3Ɛ + 1	1 + 1Ɛ + 5	2 + 6Ɛ + 6
Guo and Deng [19]	1 Ɛ	3 Ɛ	4 Ɛ	6 Ɛ	14 Ɛ
Hundera et al. [17]	3	1 + 2	4 + 2	2 + 4	10 + 8
Hussain et al. [30]	2ℌ	2ℌ	5ℌ	6ℌ	15ℌ
Ours	2ℌ	2ℌ	4ℌ	4ℌ	12ℌ

Table 3 Computational cost (millisecond)

Scheme	Warrant generation and delegation	Proxy secret key generation/warrant verification	Proxy signcryption	Unsigncryption	Total
Ming and Wang [27]	12.93	29.8	76.28	94.4	213.41
Zhou et al. [28]	12.93	34.11	29.08	76.84	152.96
Yu and Wang [29]	1.25	1.25	22.96	80.06	105.52
Guo and Deng [19]	0.97	2.91	3.88	5.82	13.58
Hundera et al. [17]	12.93	34.11	47.04	68.22	162.3
Hussain et al. [30]	0.96	0.96	2.4	2.88	7.2
Ours	0.96	0.96	1.92	1.92	5.76

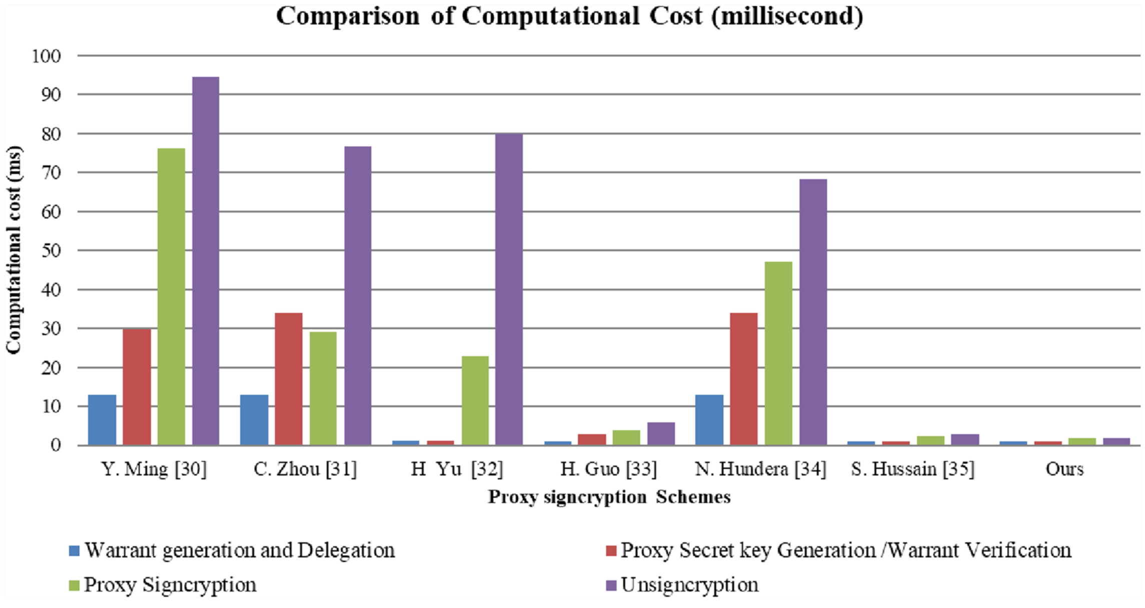


Fig. 6 Comparison of computation cost of alternative schemes with proposed HYEC-IBPSC-SGDG-FC Scheme

Table 4 Communication cost comparison

Scheme	Proxy delegation	Proxy signcryption	Total
Ming and Wang [27]	1	$1\mathbb{G}+2\mathscr{H}$	$1+1\mathbb{G}+2\mathscr{H}$
Zhou et al. [28]	$3+1\mathscr{H}$	$1+1\mathbb{G}+2\mathscr{H}$	$4+1\mathbb{G}+3\mathscr{H}$
Yu and Wang [29]	$1+1\mathscr{H}$	$1\mathbb{G}+2\mathscr{H}$	$1+1\mathbb{G}+3\mathscr{H}$
Guo and Deng [19]	1+2	2+5	3+7
Hundera et al. [17]	$+2\mathbb{G}$	$2+1\mathbb{G}+1\mathscr{H}$	$3+3\mathbb{G}+1\mathscr{H}$
Hussain et al. [30]	1+2 N	2+5 N	3+7 N
Ours	1+2 N	2+4 N	3+6 N

Table 5 Communication cost (bits)

Scheme	Proxy delegation	Proxy signcryption	Total
Ming and Wang [27]	2048	1184	3232
Zhou et al. [28]	6656	3232	9888
Yu and Wang [29]	2560	1184	3744
Guo and Deng [19]	2368	6656	9024
Hundera et al. [17]	4096	5200	9296
Hussain et al. [30]	2208	4896	7104
Ours	2208	4416	6624

Hence it can be seen that the proposed approach outperforms the alternative schemes.

6 Conclusion

For the natural gas distribution environment, we proposed the HYEC-IBPSC-SGDG-FC approach, which is both secure and efficient. In fog computing based SGDG approach, we showed that the proposed technique is able to be utilized to control data access. The privacy, authentication, integrity, and non-repudiation in our system is carried out logically in one step by using the technique of identity based proxy signcryption. As part of our formal security analysis, we proved that the proposed system is exposed to be selected plaintext attack (SPA) sheltered, assuming that the DDH assumption is hard. It is also demonstrated that the projected scheme is existential unforgeable. We also showed that the proposed technique beats the alternative schemes in terms of computing costs in milliseconds (ms) and communication cost in bits for each step of the HYEC-IBPSC-SGDG-FC scheme. The simulation study performed by utilizing AVISPA tool illustrates that HYEC-IBPSC-SGDG-FC is safe under OFMC and CL-Atse backend. The development of an attribute-based signcryption method with PRE for fine-grained access control will be the focus of our future study.

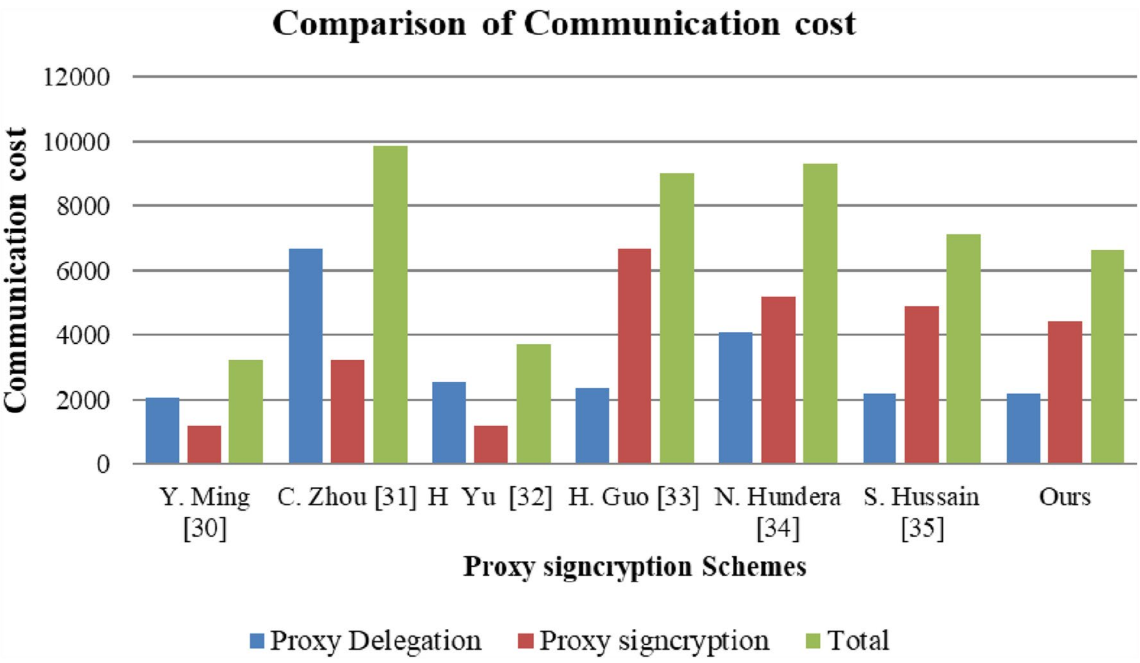


Fig. 7 Comparison of communication cost of alternative schemes with proposed HYEC-IBPSC-SGDG-FC Scheme

Declarations

Conflict of interest This research has no any declarations of interest to be disclosed.

References

1. Yeh S (2007) An empirical analysis on the adoption of alternative fuel vehicles: the case of natural gas vehicles. Energy Policy 35(11):5865–5875
2. Hackbarth A, Madlener R (2013) Consumer preferences for alternative fuel vehicles: a discrete choice analysis. Transp Res Part D: Transp Environ 25:5–17
3. Dong S, Duan S, Yang Q, Zhang J, Li G, Tao R (2017) MEMS-based smart gas metering for Internet of Things. IEEE Internet Things J 4(5):1296–1303
4. Khan MF, Zoha A, Ali RL (2007) Design and implementation of smart billing and automated meter reading system for utility gas. In: 2007 International conference on information and emerging technologies. IEEE, pp 1–6
5. Cascetta F, Vigo P (1994) The future domestic gas meter: review of current developments. Measurement 13(2):129–145
6. Wang Z, Hu C, Zheng D, Chen X (2021) Ultra-low-power sensing framework for internet of things: a smart gas meter as a case. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3110886
7. Al-Ali AR, Landolsi T, Hassan MH, Ezzeddine M, Abdelsalam M, Baseet M (2018) An IoT-based smart utility meter. In: 2018 2nd international conference on smart grid and smart cities (ICSGSC). IEEE, pp 80–83
8. Singh S, Yassine A (2018) IoT big data analytics with fog computing for household energy management in smart grids. In: International conference on smart grid and internet of things. Springer, Cham, pp 13–22
9. Bhole D, Mote A, Patil R (2016) A new security protocol using hybrid cryptography algorithms. Int J Comput Sci Eng 4(2):18–22
10. Jalasri M, Lakshmanan L (2018) A survey: integration of iot and fog computing. In: 2018 second international conference on green computing and internet of things (ICGCIoT). IEEE, pp 235–239
11. Pattewar G, Mahamuni N, Nikam H, Loka O, Patil R (2022) Management of IoT devices security using blockchain—a review. In: Sentimental analysis and deep learning, pp735–743
12. Mambo M, Usuda K, Okamoto E (1996) Proxy signatures: delegation of the power to sign messages. IEICE Trans Fundam Electron Commun Comput Sci 79(9):1338–1354
13. Li X, Chen K (2004) Identity based proxy-signcryption scheme from pairings. In: IEEE international conference on services computing, 2004.(SCC 2004). Proceedings. 2004. IEEE, pp 494–497
14. Wang Q, Cao Z (2005) Efficient ID-based proxy signature and proxy signcryption form bilinear pairings. In: International conference on computational and information science. Springer, Berlin, Heidelberg, pp. 167–172
15. Swapna G, Gopal PVSSN, Gowri T, Reddy PV (2012) An efficient ID-based proxy signcryption scheme. Int J Inf Netw Secur 1(3):200
16. Yu H, Wang Z, Li J, Gao X (2018) Identity-based proxy signcryption protocol with universal composability. Secur Commun Netw 2018:1–11
17. Hundera NW, Mei Q, Xiong H, Geressu DM (2020) A secure and efficient identity-based proxy signcryption in cloud data sharing. KSII Trans Internet Inf Syst (TIIS) 14(1):455–472
18. Von Oheimb D (2005) The high-level protocol specification language HLPSL developed in the EU project AVISPA. In: Proceedings of APPSEM 2005 workshop, pp 1–17
19. Guo H, Deng L (2020) An identity based proxy signcryption scheme without pairings. Int J Netw Secur 22(4):561–568
20. Yang X, Xi W, Ren N, Wang J, Li M (2021) Support outsourcing unsigncryption and member revocation identity-based proxy signcryption scheme with drone environment. J Phys: Conf Ser 1828(1):012119
21. Waheed A, Umar AI, Zareei M, Din N, Amin NU, Iqbal J, Saeed Y, Mohamed EM (2020) Cryptanalysis and improvement of a proxy signcryption scheme in the standard computational model. IEEE Access 8:131188–131201
22. Khan A, Ullah I, Algarni F, Naeem M, Uddin MI, Khan MA (2022) An efficient proxy blind signcryption scheme for IoT. CMC-Comput Mater Continua 70(3):4293–4306
23. Paterson KG (2002) ID-based signatures from pairings on elliptic curves. IEEE Commun Lett 38(18):1025–1026
24. Yu Y, Yang B, Sun Y, Zhu S (2009) Identity based signcryption scheme without random oracles. Comput Stand Interfaces 31(1):56–62
25. Yogesh PR (2020) Formal verification of secure evidence collection protocol using BAN logic and AVISPA. Proc Comput Sci 167:1334–1344
26. Patil RY, Devane SR (2019) Network forensic investigation protocol to identify true origin of cyber crime. J King Saud Univ-Comput Inf Sci
27. Ming Y, Wang Y (2015) Proxy signcryption scheme in the standard model. Secur Commun Netw 8(8):1431–1446
28. Zhou C, Zhang Y, Wang L (2018) A provable secure identity-based generalized proxy signcryption scheme. Int J Netw Secur 20(6):1183–1193
29. Yu H, Wang Z (2019) Construction of certificateless proxy signcryption scheme from CMGs. IEEE Access 7:141910–141919
30. Hussain S, Ullah I, Khattak H, Khan MA, Chen CM, Kumari S (2021) A lightweight and provable secure identity-based generalized proxy signcryption (IBGPS) scheme for Industrial Internet of Things (IIoT). J Inf Secur Appl 58:102625

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.