IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 23, 2019, accepted December 11, 2019, date of publication December 17, 2019,
date of current version December 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960291

An Efficient and Lightweight Deniably Authenticated Encryption Scheme for e-Mail Security

JAYAPRAKASH KAR[1], KSHIRASAGAR NAIK[2], AND TAMER ABDELKADER[3]

[1]Centre for Cryptography, Cyber Security and Digital Forensics, Department of Computer Science & Engineering, The LNM Institute of Information Technology, Jaipur 302031, India
[2]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[3]Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt

Corresponding author: Jayaprakash Kar (jayaprakashkar@lnmiit.ac.in)

ABSTRACT The most important security requirements to secure electronic mail (e-mail) systems are: confidentiality, authentication, non-repudiation and data integrity. In conventional e-mail systems, Secure/Multipurpose Internet Mail Extensions (S/MIME) and Pretty Good Privacy (PGP) digital envelops are used to satisfy these security requirements. However, confidentiality and authentication are performed in two different phases, which increases computations and leads to more energy consumption. Moreover, the receiver can easily reveal the source of the message, violating the sender's privacy. In this paper, we propose a low-cost deniably authenticated encryption scheme (**DA-ENS**), where all the cryptographic primitives are being performed in a single logical step to achieve these goals. Experimental results show that our scheme, **DA-ENS**, achieves low computational cost and communication overhead at 80-bit, 112-bit, 128-bit, 192-bit and 256-bit security levels. Energy consumption is shown to be reduced to 80%, 67%, 42%, 62% and 48% compared to similar schemes SL+BF, LXJ+BF, Fagen Li *et al.*(FL), AJL and CZJZJSZ respectively. Also, we have proven that, our scheme **DA-ENS** is provably secure in random oracle model.

INDEX TERMS e-mail security, data integrity, non-repudiation, deniable authentication.

I. INTRODUCTION

In the current era of digital information, electronic mail (e-mail) is considered to be a common and widely used medium of communication. Whether it is used for personal or general purposes, business or non-business purposes, security is an essential and critical requirement. An e-mail systems is vulnerable to several security threats. These threats are summarized in the following list:

- Privacy Invasion: When all or part of a message is revealed to an unauthorized person.
- Theft of identity: When a person impersonates another one, thereby reading and sending e-mails as if he is the true owner of the e-mail account.
- Message tampering: An unauthorized person intercepts a message and alters its contents.
- Repudiation: The denial of the message origin or its authenticity.

The associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg.

To protect e-mail systems against the prospective threats, a secure e-mail system should satisfy the following security requirements [1]–[3]:

- Confidentiality: The message is read by the intended receiver(s) only.
- Authenticity: The message originated from the designated sender. The recipient should be able to prove that the given e-mail is really initiated by the specified sender.
- Data Integrity: This ensures that the message has not been changed, by an unauthorized user, before or during the transmission of the message.
- Strong fairness [1]: The sender should obtain a proof of receipt from the receiver, when the receiver receives the e-mail certified by the sender. This process protects from false denials.

There are three major types of e-mail encryption protocols that can help in satisfying the above security requirements: Simple Mail Transfer Protocol(SMTP) [4],

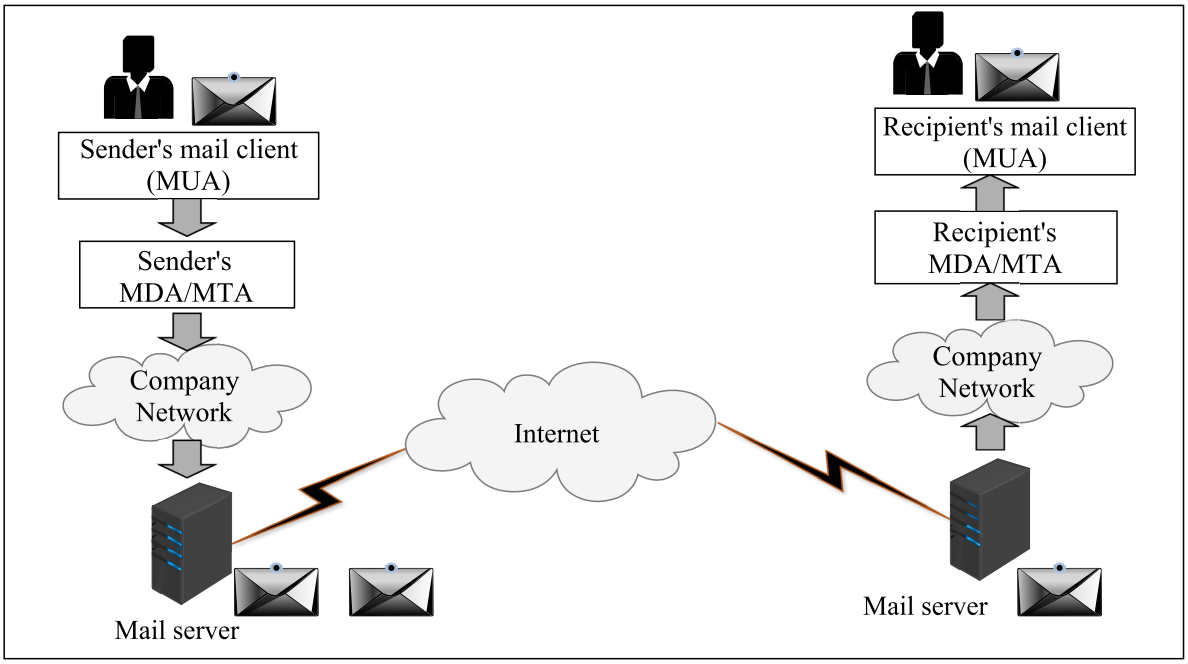

FIGURE 1. e-mail workflow model.

Secure/Multipurpose Internet Mail Extension (S/MIME) [5] and Pretty Good Privacy (PGP) [6]. The drawback of SMTP is that a person with administrative privileges to SMTP servers can modify or even delete the e-mail sent by other people through the SMTP servers. In PGP, the main drawback is in the key distribution mechanism where PGP key servers are used. Anyone can construct and use a key having any name on it [7], [8]. For example, there may be numerous keys on the PGP key servers with the name "Bill Gates" on them, but none of them may actually belong to the founder of Microsoft Corporation.

A. WORKFLOW OF e-MAIL SYSTEMS

A standard e-mail system model is presented in Fig 1. An e-mail message is sent from the sender's Mail User Agent (MUA), such as Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredible or Lotus Notes, to the mail server that contains a Mail Transfer Agent (MTA) software. A sender MTA transfers the message to the recipient's MTA using the Simple Mail Transfer Protocol (SMTP), so they are logically called SMTP servers. The recipient's MTA then delivers the message to the incoming Mail Delivery Agent (MDA), which stores the message and waits for the recipient to read it. There are two main protocols used to retrieve e-mails from an MDA: POP3 (Post Office Protocol), and IMAP (Internet Message Access Protocol). POP3, the older of the two, is used to retrieve e-mails and, in certain cases, leave a copy of it on the server. IMAP is used to coordinate the e-mail status (read, deleted, moved) across multiple e-mail clients. With IMAP, a copy of every message is stored in the server, and is synced with all the copies in the clients. For this reason, incoming mail servers are called POP servers or IMAP servers, depending on which protocol is used. To protect e-mails from unauthorized access, MDA is protected by a user login-name and password.

B. MOTIVATION AND CONTRIBUTION

The two most important security requirements for a secure e-mail system are confidentiality and authentication. PGP and S/MIME are well-known e-mail protocols that provide the two security goals by performing encryption and digital signature in two individual respective phases. However, the two protocols operate as digital envelopes, which results in heavy computational cost in terms of processing time and energy consumption. One of the main drawbacks of traditional e-mail systems is that a recipient, after receiving the intended message, may easily disclose the source of the message, violating the sender's privacy. This means it lacks deniable authentication. In [9], a deniable authenticated encryption scheme for e-mail systems is proposed. However this scheme is interactive and based on pairing, which renders the computational cost significantly high.

In this paper, we propose our design of a provably secure deniable authenticated encryption scheme **DA-ENS**, where all security goals, confidentiality, non-repudiation, integrity and deniable authentication, can be achieved in a single logical step. Since the pairing operation is a heavy cryptographic operation, we have avoided using it in our scheme. From the empirical analysis, We have shown that the proposed scheme **DA-ENS** achieves low computational cost, communication overhead and energy consumption. Moreover, we compare the performance using different security levels: 80-bit, 112-bit, 128-bit, 192-bit and 256 bit.

The rest of the paper is organized as follows. A brief overview of related work is presented in Section II. In Section III, we introduce and explain the mathematical assumptions related to our work. The adversary model is presented in Section IV-A, and the proposed scheme in Section V. The performance analysis is conducted in Section VI. Experiments and their results are discussed and analyzed in Section VII. Finally, conclusions are drawn in Section VIII.

II. RELATED WORK

In this section, we present the previous work related to secure e-mail protocols PGP, S/MIME, deniable authentication(DA) and authenticated encryption(AE). Authentication can be classified into two types: entity authentication and message authentication. Entity authentication is the process whereby one party is assured, through acquisition of corroborative evidence, of the identity of a second party involved in a protocol,and that the second has actually participated. Message or data authentication is a procedure that allows communicating parties to verify that the received or stored messages are authentic. An authenticated Encryption (AE) scheme should achieve two security goals confidentiality and authenticity. AE is the set of cryptographic primitives that allow the communicating parties to verify that the received ciphertext is authentic. An AE scheme is designed either based on a symmetric key cryptosystem or an asymmetric-key cryptosystem [10]–[12].

Deniable Authentication (DA) is different from traditional authentication cryotographic. In deniable authentication, the intended receiver can identify the source of the given message but cannot prove the source of the message to a third party. There are many applications that require DA, such as electronic voting, secure negotiation over internet, and privacy-preserving location-based services [13]–[15]. The symmetric key based AE is deniable, whereas the asymmetric key based AE is non-deniable.

Security in e-mail applications was not in the focus in the early days of the Internet. e-mail application architectures were more interested in improving the transmission and reception operations than in studying security gaps. One of the early security protocols used in e-mail applications is the public-key cryptography. In this protocol, each user has a pair of keys, public and private. A user, Bob, publishes his public key to the others so that they can use it to encrypt the messages sent to Bob. When Bob receives an encrypted message, he uses his private key to decrypt the message. The two well-known protocols, S/MIME [16] and PGP [17] were first introduced for e-mail security. Subsequently, a few other early-stage protocols, namely GPG [18], [19] and Transport Layer Security (TLS) [18], [20], have emerged. Some e-mail providers, such as Google in its Gmail application, use TLS, which evolved from the Secure Socket Layer (SSL) protocol [18].

In PGP and S/MIME, each user has to maintain *two* pairs of public/private keys. One pair is used for message encryption, and the other pair is used for digital signature. Both PGP and S/MIME use digital envelopes to provide message confidentiality. The operation of these protocols is described as follows:

- Sender, Bob, randomly picks a number, and use it as a session key to encrypt the message by using symmetric key encryption scheme.
- Using asymmetric key cryptosystem, Bob encrypts the session key using the receiver's public key.
- The encrypted message and the encrypted session key are concatenated together and sent to the receiver, Alice.
- After receiving the encrypted message, Alice decrypts it using her private key to get the session key.
- Alice uses the session key to decrypt the cipher text and obtain the original message.

Both protocols use digital signature for message authentication, which is described as follows:

- The sender, Bob, signs the message digest using his private key.
- The resulting signature is attached to the encrypted message.
- The receiver, Alice, verifies the validity of the signature using Bob's public key.

Since digital signatures provide non-repudiation evidence of the sender, the receiver can prove the source to any third party. To resolve this problem, Harn and Re [2] constructed a new scheme, namely the HR scheme, to provide deniable authentication in e-mail systems. In the HR scheme, a sender signs the ciphertext of a session key directly instead of signing the message digest, which makes the signature forgeable to achieve deniability for the authentication. In this construction, the recipient can identify the origin of the given message, but it cannot prove the source to any third party. Hence, this provides deniable authentication. However, Ki *et al.* [21] proved that the HR scheme is not fully deniable. The transcripts generated by the sender are realistically distinguishable from those generated by a receiver when the public key encryption scheme is secure against chosen cipher text attack (CCA).

A fully deniable authentication scheme, called HLLC, was proposed by Harn *et al.* [22] in 2011. Although the scheme provides confidentiality, it is not suitable for e-mail security because of its interactive nature. In addition, the authors didn't provide a formal security proof of the scheme. Similarly, there have been several deniable authentication schemes [23] that are not suitable for e-mail systems, because of their interactive natures.

III. MATHEMATICAL ASSUMPTIONS

Public key cryptography relies upon some mathematical hard problems such as Integer Factorization Problem (IFP), Discrete Logarithm Problem (DLP), Diffie-Hellman Problem (DHP). In this paper, we consider the discrete logarithm and Diffie-Hellman problem based on Elliptic Curve. The elliptic curve discrete logarithm problem is defined as follows:

Definition 1: Let E be an elliptic curve over a finite field $\mathbb{F}_q$, where q is a prime number. Given two points P, $Q \in E(\mathbb{F}_q)$ such that $Q \in \langle P \rangle$, compute d such that $Q = [d]P$, where d is an integer.

Definition 2: The Discrete Logarithm (DL) problem is hard; Let $\mathbb{G}$ be an additive cyclic group with generator P. The assumption (t, ϵ)-DL holds in $\mathbb{G}$ if there does not exist any adversary A with running time t that has advantage ϵ in solving the DL problem.

To achieve confidentiality over an insecure channel, the Elliptic-Curve Diffie-Hellman (ECDH) protocol is used. The ECDH allows two parties, having an elliptic-curve public/private key pair, to generate a symmetric secret key, and use it to encrypt subsequent messages. The ECDH allows the shared secret key to be computed independently by each party, using some agreed upon parameters, and the public/private keys. The ECDH or Computational Diffie-Hellman (CDH) problem is defined as follows:

Definition 3: Let $\mathbb{G}$ be an additive cyclic group with generator P and order n. Let $a, b \in \mathbb{Z}_q^*$ be chosen randomly in the interval $[1, n-1]$. Then the shared secret key is computed using the product of abP.

Definition 4: The Computational Diffie-Hellman problem (CDH) is hard: Let $\mathbb{G}$ be an additive cyclic group with generator P. The assumption (t, ϵ)-CDH in $\mathbb{G}$ holds, if there does not exist any polynomially solvable adversary A in running time t which has advantage ϵ in solving the CDH problem.

Given an additive cyclic group $\mathbb{G}$ formed on the elliptic curve $\mathbb{E}$ over finite field $\mathbb{F}_q$, denoted by $E(\mathbb{F}_q)$. The cyclic group $\mathbb{G}$ is of prime order q and primitive element P. The Decisional Diffie-Hellman (DDH) problem is defined as follows:

Definition 5: Given the elements (P, aP, bP, cP) for unknown $a, b, c \in \mathbb{Z}_q^*$, decide whether $c \equiv ab \bmod q$ holds, then (P, aP, bP, cP) is called a valid Diffie-Helman tuple. In the complexity assumption, there is a well-known problem, known as Gap Diffie-Hellman (GDH) problem. The GDH problem is to compute a given instance (P, aP, bP) of the CDH problem applying the DDH oracle. Given the instances (P, aP, bP, cP), decide whether $c \equiv ab \bmod q$ or not. If (P, aP, bP, cP) is a valid Diffie-Hellman tuple, we denote $DDH(P, aP, bP, cP) =$ "1"; otherwise $DDH(P, aP, bP, cP) =$ "0".

Definition 6: The $(\epsilon_{gdh}, t, q_{ddh})$-GDH assumption holds if there does not exist a polynomially solvable adversary $\mathcal{A}$ that have an advantage of at least ϵ_{gdh} to solve the GDH problem after the submission of at most q_{ddh} number of queries to the DDH oracle in t running time.

IV. A GENERIC FRAMEWORK OF THE DENIABLE AUTHENTICATED ENCRYPTION (DAE) SCHEME

In this section, we define a generic model of the Deniably Authenticated Encryption (DAE) scheme, which consists of the following four probabilistic polynomially solvable algorithms.

- `Setup`: Given a security parameter θ as input, it generates the system parameter *param* as output.
- `KeyGen`: The key generation algorithm takes the system parameter *param* as input, and returns the public and the private key pairs, (PK_s, SK_s) and (PK_r, SK_r), of the sender and the receiver, respectively.
- `DAuth-Encrypt`: The deniably authenticated encryption algorithm takes as inputs the message m, the sender's private key SK_s, the sender's public PK_s, and the receiver's public key PK_r, and produces the ciphertext σ.
- `DAuth-Decrypt`: The deniably authenticated decryption algorithm takes as inputs the cipher text σ, the receiver's private key SK_r, the receiver's public PK_s and the sender's public key PK_r. If σ is an invalid ciphertext, it outputs an error symbol $\perp$, otherwise it returns the plain text message m.

The model is represented in Figure 2 for better understanding.

A. ADVERSARY MODEL AND SECURITY

The two important security goals, namely confidentiality and deniable authentication, are achieved by a DAE algorithm to be presented in algorithm-2. For confidentiality, the security notion is indistinguishable against adaptive Chosen Ciphertext Attack (IND-CCA2) and deniably authenticated against chosen message attack (DA-CMA) [24].

B. SECURITY NOTION FOR IND-CCA2

We adopt this notion where the IND-CCA2 game is played between the challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ such that $\mathcal{B}$ uses $\mathcal{A}$ as a subroutine. The notion is described as follows:

`Initial`: $\mathcal{B}$ executes the **Setup** algorithm taking the security parameter, θ, as input and outputs the systems parameter, *param*.

`KeyGen`: The systems parameters, *param*, are passed to the **KeyGen** algorithm in $\mathcal{B}$ to obtain the public and the private key pairs (PK_s, SK_s) and (PK_r, SK_r) for both the sender and the receiver, respectively. Then $\mathcal{B}$ provides the public keys PK_s and PK_r to $\mathcal{A}$. Now, the game undergoes the following two phases:

`Phase-I`: $\mathcal{A}$ can ask a series of polynomially bounded number of queries to the deniably authenticated encryption and decryption oracles in a adaptive manner. $\mathcal{A}$ provides a message m to $\mathcal{B}$, and $\mathcal{B}$ executes the deniably authenticated encryption oracle which outputs the ciphertext, σ, as shown in Equation 1

$$\texttt{DAuth} - \texttt{Encrypt}(m, SK_s, PK_s, PK_r) = \sigma \qquad (1)$$

Then $\mathcal{B}$ sends σ to $\mathcal{A}$. In deniably authenticated decryption, oracle $\mathcal{A}$ submits a ciphertext σ to $\mathcal{B}$. Then $\mathcal{B}$ executes the oracle, which outputs the plain text message m provided that the ciphertext is valid one. Otherwise, it returns an error

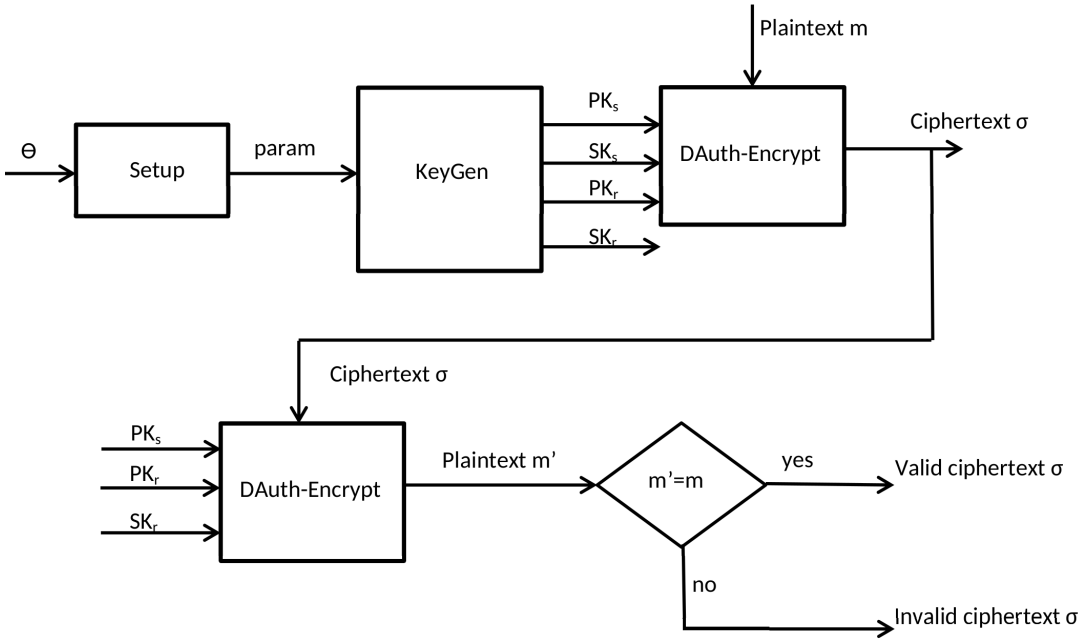

FIGURE 2. A generic framework of the Deniably Authenticated Encryption (DAE) scheme.

symbol $\perp$ to $\mathcal{A}$, as shown in Equation 2

$$\texttt{DAuth-Encrypt}(\sigma, SK_r, PK_r, PK_s) = \begin{cases} m & \text{if m is valid,} \\ \perp & \text{if m not valid} \end{cases} \tag{2}$$

`Challenge`: $\mathcal{A}$ selects two plaintext messages m_0 and m_1 of equal length and sends them to $\mathcal{B}$. $\mathcal{B}$ picks a random bit $\xi \in \{0, 1\}$ and executes the oracle of deniably authenticated encryption on m_ξ that outputs the ciphertext $\tilde{\sigma}$. This is obtained from the oracle $\texttt{DAuth-Encrypt}(m_\xi, SK_s, PK_s, PK_r)$. $\mathcal{B}$ sends the challenge ciphertext $\tilde{\sigma}$ to $\mathcal{A}$.

`Phase-II`: As in phase-I, $\mathcal{A}$ would submit a series of polynomially bounded number of queries in an adaptive manner to the deniably authenticated encryption and decryption oracles. Here the limitation is that $\mathcal{A}$ cannot send queries to deniably authenticated decryption oracle on the challenge ciphertext .

`Guess`: $\mathcal{A}$ produces ξ' as its guess, and wins the game if $\xi' = \xi$. The advantage of $\mathcal{A}$ can be defined as in Equation 3.

$$Adv_{\text{DAE}}^{\text{IND-CCA2}}(\mathcal{A}) = 2Pr[\xi' = \xi] - 1 \tag{3}$$

C. SECURITY NOTION FOR DA-CMA

The notion of security would be specified as deniable authentication against chosen message attack (DA-CMA), following the security notion described in [24]. In DAE, both the sender and the receiver would produce a valid ciphertext. In digital signature, only the sender can produce a valid signature using his own private key. The accepted security notion for digital signature is existential unforgeability against chosen message attack (EUF-CMA) in an adaptive manner. A modified version of EUF-CMA in DAE is called DA-CMA. The DA-CMA game, played between a challenger $\mathcal{B}$ and an adversary $\mathcal{F}$, can be described as follows:

`Initial`: $\mathcal{B}$ executes the **Setup** algorithm taking the security parameter, θ, as input, and producing the system parameter *param* as output.

`KeyGen`: The **KeyGen** algorithm in $\mathcal{B}$ takes *param* as input, and provides the public and private key pairs (PK_s, SK_s) and (PK_r, SK_r) for both the sender and the receiver respectively. Then $\mathcal{B}$ provides the public keys PK_s and PK_r to $\mathcal{F}$.

`Attack`: $\mathcal{F}$ would ask a polynomially bounded number of queries to the deniably authenticated encryption and decryption oracles in the DA-CMA game. At the end of the game, $\mathcal{F}$ produces a valid ciphertext $\tilde{\sigma}$, and wins the game if and only if the following conditions hold:

1) $\texttt{DAuth-Decrypt}(\sigma^*, SK_r, PK_r, PK_s) = m^*$
2) $\mathcal{F}$ has not submitted a query to deniably authenticated encryption oracle on m^*

`Guess`: $\mathcal{F}$ produces ξ' as its guess, and wins the game if $\xi' = \xi$. The advantage of $\mathcal{F}$ can be defined as in Equation 4.

$$Adv_{\text{DAE}}^{\text{DA-CMA}}(\mathcal{F}) = 2Pr[\xi' = \xi] - 1 \tag{4}$$

V. THE PROPOSED SCHEME

Based on the generic model defined in Section IV, our scheme consists of three main algorithms namely, key generation **KeyGen**, Deniable Authenticated Encryption **DAuth-Encrypt** and Deniable Authenticated Decryption **DAuth-Decrypt**, in addition to a fourth one to check deniability **Deniability-Check**.

In the construction of the protocol **DA-ENS**, the system sets an elliptic curve E over $\mathbb{Z}_p$ such that $E(\mathbb{Z}_p)$ forms an abelian group. Further, the number of points in $E(\mathbb{Z}_p)$ is

divisible by a large prime, say n. Let $P \in E(\mathbb{Z}_p)$ be the generator, or base point of order n. The system chooses an integer θ as a security parameter such that $n > 2^{\theta}$. It sets two collision resistant hash functions H_1 and H_2, such that $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}$. This publishes the systems parameters *param* which is (n, E, p, P, H_1, H_2).

Both the sender and the receiver key pairs are generated by key generation algorithm, **KeyGen**, as shown in Algorithm 1.

Algorithm 1 KeyGen

INPUT: $(P, \mathbb{G}, q, \theta)$
OUTPUT: Private and public key pairs (d_s, Q_s) for sender and (d_r, Q_r) for receiver.
begin
    Randomly choose d_s and $d_r \in [1, n-1]$.
    $Q_s = d_s \cdot P$ and $Q_r = d_r \cdot P$
    return (d_s, Q_s) and (d_r, Q_r).
end

Deniably authenticated encryption is performed by generating ciphertext and deniable authentication in one single logical step. To resist the **Replay attack**, sender concanates the message m along witht the timestamp T and construct a new message $m^* = m \| T$. All computations for these cryptographic primitives are defined in **DAuth-Encrypt**, as shown in Algorithm 2.

Algorithm 2 `DAuth-Encrypt`

INPUT: (m^*, Q_r, Q_s, P)
OUTPUT: $\sigma \leftarrow (c, \lambda, U, R)$
begin
    $u \in [1, n-1]$ is chosen randomly.
    $W \leftarrow u \cdot Q_r, k \leftarrow H_1(W)$
    $c \leftarrow m^* \oplus k$
    $\lambda \leftarrow H_2(m^* \| W \| Q_s \| Q_r)$
    $\gamma \leftarrow (u + \lambda \cdot d_s) \bmod n$
    $U \leftarrow \gamma \cdot P, R \leftarrow \gamma \cdot Q_r$
    return $\sigma \leftarrow (c, \lambda, U, R)$
end

The output of **DAuth-Encrypt** are the tuples (c, λ, U, R) which represent the deniably authenticated ciphertext. When the recipient receives σ, as shown in Algorithm 2, it performs exclusive OR operation bit-by-bit on ciphertext c and obtains the plaintext message m^*. Finally, the receiver verifies the resulting message. If it is correct, it accepts the message, otherwise returns a symbol "$\perp$" for rejection.

In Algorithm-2, the encryption of the message is performed using an exclusive OR operation bit-by-bit. We can consider the symmetric encryption scheme, Advanced Encryption Standard (AES), which encrypts the message m into c, $E_k(m^*) = c$, where $c \leftarrow m^* \oplus k$, and the decryption will be $D_k(c) = m^*$, where $m^* \leftarrow c \oplus k$. The symmetric encryption scheme is considered to protect the proposed scheme against passive attacks [9] and to make the encryption and decryption process faster.

Operation of the proposed scheme **DA-ENS** described in algorithms 2 and 3 is represented in Figure 3. The symbol **OP** is used to denote the cryptographic operations; addition, multiplication and subtraction. Other symbols are defined in Table 1.

Algorithm 3 `DAuth-Decrypt`

INPUT: (σ, Q_r, d_r, Q_s)
OUTPUT: m^*
begin
    $W \leftarrow (U - \lambda \cdot Q_s) d_r$
    $k \leftarrow H_1(W)$
    $m^* \leftarrow c \oplus k$
    if $(\lambda = H_2(m^* \| W \| Q_s \| Q_r)$ and $d_r U = R)$
        m^*
    else return $\perp$
end

TABLE 1. Nomenclature.

Symbol/Notation	Description
E	Elliptic curve
$\mathbb{Z}_p$	integer modulo p $i.e$ the set $\{0, 1 \ldots p-1\}$
$\mathbb{Z}_q^*$	multiplicative group of $\mathbb{Z}_q$ $i.e\ \mathbb{Z}_q^* = \{a \mid 1 \leq a \leq q-1\}$, q is prime.
n	large prime that divides the number of points in $E(\mathbb{Z}_p)$
$\mathbb{G}$	cyclic additive group of order n
m	plain text message
c	ciphertext
θ	security parameter
P	Base point of the group $\mathbb{G}$
d_s, Q_s	Sender's private and public key
d_r, Q_r	Receiver's private and public key
H_1, H_2	Collision resistant hash function
$\perp$	Null value.
$\|\|$	Concatenation operation (concatenate two strings
$\oplus$	Exclusive OR operation

A. PROOF OF CORRECTNESS AND DENIABILITY

This section shows the consistency of the elements W and R. It follows that:

$$\begin{aligned}(U - \lambda \cdot Q_s) d_r &= (\gamma \cdot P - \lambda \cdot d_s P) d_r \\ &= d_r(\gamma - \lambda d_s) \cdot P \\ &= u \cdot d_r P = u \cdot Q_r = W\end{aligned}$$

In addition, we need to show that $d_r U = R$. It follows that:

$$d_r U = d_r \gamma P = \gamma Q_r = R$$

Definition 7: An encryption scheme is said to be deniable if both ciphers generated from the same message, by the sender and the receiver, using their own private keys, are indistinguishable from each other.

The ciphertext generated by the receiver using its private key d_r would be indistinguishable from the ciphertext generated by the sender with its private key d_s. To simulate

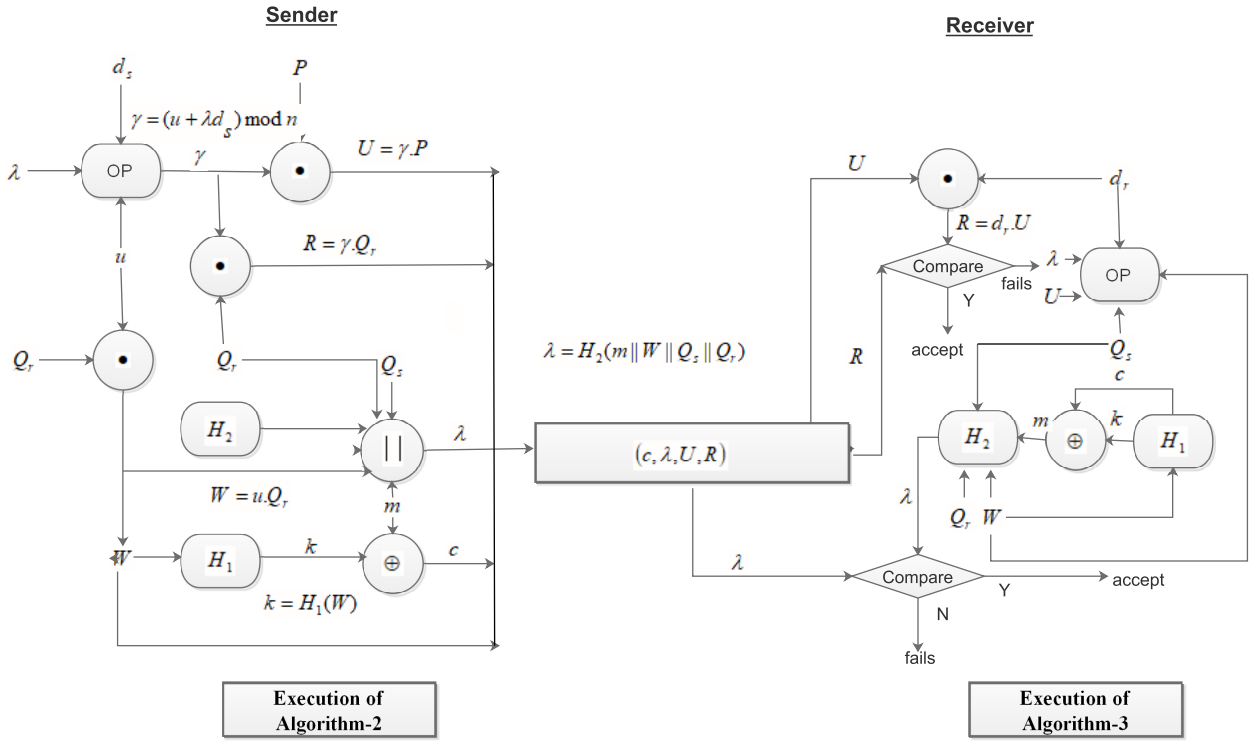

FIGURE 3. Operation of DA-ENS.

the transcripts $(\tilde{c}, \tilde{\lambda}, \tilde{U}, \tilde{R})$ generated for the message m^*, the receiver perform the deniability algorithm, as shown in Algorithm 4.

Algorithm 4 Deniability-Check

1: INPUT: (m^*, Q_s, Q_r)
2: OUTPUT: $(\tilde{c}, \tilde{\lambda}, \tilde{U}, \tilde{R})$
3: **begin**
4: $\tilde{u} \in [1, n-1]$ randomly chosen.
5: $\tilde{W} \leftarrow \tilde{u} \cdot P, \tilde{k} \leftarrow H_1(\tilde{W})$
6: $\tilde{c} \leftarrow m^* \oplus \tilde{k}$
7: $\tilde{\lambda} \leftarrow H_2(m^* \| \tilde{W} \| Q_s \| Q_r)$
8: $\tilde{U} \leftarrow \lambda \tilde{Q}_s + \tilde{u}P$ and $\tilde{R} \leftarrow d_r \cdot \tilde{U}$
9: **return** $\tilde{\sigma} \leftarrow (\tilde{c}, \tilde{\lambda}, \tilde{U}, \tilde{R})$
10: **end**

Let $(\hat{c}, \hat{\lambda}, \hat{U}, \hat{R})$ be a ciphertext that is chosen randomly in the set of valid ciphertext generated by the sender. Let $\mathcal{B}$ intend the ciphertext $(\hat{c}, \hat{\lambda}, \hat{U}, \hat{R})$ to receiver. The probability

$$Pr[(\tilde{c}, \tilde{\lambda}, \tilde{U}, \tilde{R}) = (\hat{c}, \hat{\lambda}, \hat{U}, \hat{R})] = \frac{1}{1-n}.$$

because $(\tilde{c}, \tilde{\lambda}, \tilde{U}, \tilde{R})$ is constructed from a randomly chosen value $\tilde{u} \in [1, n-1]$.

B. SECURITY

Security of the proposed scheme **DA-ENS** relies on the mathematical hard problems **CDH** and **GDH** defined in Section-III. The security notion for confidentiality is indistinguishability against chosen ciphertext attack, and for digital signature is existential forgeability against chosen message attack. Both security notions have been proven in theorems 1 and 2 respectively.

Theorem 1: Assume that there is an IND-CCA2 adversary $\mathcal{A}$ which can distinguish the ciphertext during the IND-CCA2 game with an advantage ϵ against ***DA-ENS*** *running in time t. The adversary $\mathcal{A}$ asks at most q_{h_1} and q_{h_2} queries to random oracles H_1 and H_2, respectively, and q_e and q_d queries to deniable authenticated encryption and decryption oracles, respectively. Then there exists an algorithm $\mathcal{B}$ that can solve GDH problem in time t' and q_{ddh} queries with probability*

$$\epsilon_{dae} \leq \epsilon_{gdh} + \frac{q_e(q_{h_1} + q_{h_2}) + q_d}{2^\theta}$$

where $t' = \mathcal{O}(t + t_{h_1} + t_{h_2} + t_e + t_d)$ and $q_{ddh} = \mathcal{O}(q_{h_1} + q_{h_2} + q_d)$

Proof: See Appendix A. □

Theorem 2: Assume that there is an DA-CMA adversary $\mathcal{F}$ which is able to forge a given ciphertext during the DA-CMA game with an advantage ϵ against ***DA-ENS*** *running in time t. The adversary $\mathcal{F}$ asks at most q_{h_1} and q_{h_2} queries to random oracles H_1 and H_2 respectively and q_e and q_d queries to deniable authenticated encryption and decryption oracles respectively. Then there exists an algorithm $\mathcal{B}$ that is able to solve the GDH problem in time t' and using q_{ddh} queries with probability*

$$\epsilon_{dae} \leq \epsilon_{gdh} + \frac{q_e(q_{h_1} + q_{h_2}) + q_d + 1}{2^\theta}$$

TABLE 2. Notations used in the performance analysis.

Notation	Description
$Pairing$	The pairing operation performed
$Mul(\mathbb{G}_1)$	The point multiplication in the adaptive group $\mathbb{G}_1$
$Exp(\mathbb{G}_2)$	The exponent operation in the multiplicative group $\mathbb{G}_2$

TABLE 3. Communication overhead at different security levels.

Scheme	Security Level	Communication Overhead
SL+BF	80-bit	$235 + \|m\|/4$ bytes
	112-bit	$440 + \|m\|/4$ bytes
	128-bit	$640 + \|m\|/4$ bytes
	192-bit	$1530 + \|m\|/4$ bytes
	256-bit	$3136 + \|m\|/4$ bytes
LXJ+BF	80-bit	$258 + \|m\|/4$bytes
	112-bit	$512 + \|m\|/4$ bytes
	128-bit	$768 + \|m\|/4$ bytes
	192-bit	$1920 + \|m\|/4$ bytes
	256-bit	$3840 + \|m\|/4$ bytes
FL	80-bit	$193 + \|m\|/8$bytes
	112-bit	$384 + \|m\|/8$ bytes
	128-bit	$576 + \|m\|/8$ bytes
	192-bit	$1440 + \|m\|/8$ bytes
	256-bit	$2880 + \|m\|/8$ bytes
AJL	80-bit	$193 + \|m\|/4$ bytes
	112-bit	$384 + \|m\|/4$ bytes
	128-bit	$576 + \|m\|/4$ bytes
	192-bit	$1440 + \|m\|/4$ bytes
	256-bit	$2880 + \|m\|/4$ bytes
CZJZJSZ	80-bit	$193 + \|m\|/8$ bytes
	112-bit	$384 + \|m\|/8$ bytes
	128-bit	$576 + \|m\|/8$ bytes
	192-bit	$1440 + \|m\|/8$ bytes
	256-bit	$2880 + \|m\|/8$ bytes
DA-ENS	80-bit	$150 + \|m\|/4$bytes
	112-bit	$284 + \|m\|/8$ bytes
	128-bit	$416 + \|m\|/8$ bytes
	192-bit	$1008 + \|m\|/8$ bytes
	256-bit	$1984 + \|m\|/8$ bytes

where $t' = \mathcal{O}(t + t_{h_1} + t_{h_2} + t_e + t_d)$ and $q_{ddh} = \mathcal{O}(q_{h_1} + q_{h_2} + q_d)$.

Proof: See Appendix B □

VI. PERFORMANCE ANALYSIS

In this section, we compute the computational cost in terms of the cryptographic operations executed in a scheme. In addition, we evaluate the computational time and communication overhead with respect to the security levels: 80-bit, 112-bit, 128-bit, 192-bit and 256-bit [25], and compare with the related works [9]. The schemes in [26], [27] and [10] are not directly deniable-authenticated encryption schemes. Rather, they provide deniability of authenticated messages followed by encryption.

The schemes, SL+BF and LXJ+BF, are named following the notation used in [9]. The SL+BF scheme combines the deniable authentication scheme proposed in [26] and the encryption scheme provided in [27]. Similarly, the LXJ+BF scheme combines the work proposed in [10] for deniable authentication and the encryption scheme in [26]. AJL and CZJZJSZ are the Certificateless deniably authenticated encryption schemes proposed in [28] and [29] respectively.

TABLE 4. Size of group element and compressed size.

Security level	$\|\mathbb{G}_1\|$(bits)	compressed size(bytes)	$\|\mathbb{G}_2\|$(bits)	compressed size(bytes)
80-bit	1024	65	2048	128
112-bit	2048	128	4096	256
128-bit	3072	192	6144	384
192-bit	7680	480	15630	960
256-bit	1530	960	3072	1920

TABLE 5. Security level specification in bits.

Security level	Size of p	Size of q
80-bit	512	160
112-bit	1024	224
128-bit	1536	256
192-bit	3840	384
256-bit	7680	512

TABLE 6. Size of group element in bits.

Security level	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_2\|$	$\|MAC\|$
80-bit	1024	2048	160
112-bit	2048	4096	224
128-bit	3072	6144	256
192-bit	7680	15360	384
256-bit	15360	30720	512

Since paring is a costly cryptographic operation, our proposed scheme **DA-ENS** avoids using it. The computational cost is evaluated in terms of cryptographic operations performed at both the sender and the receiver end. The cost has been compared with the related schemes and illustrated in Table 7. Communication overhead is computed in terms of group element size, length of message digest, and size of messages. The communication overhead is evaluated and compared with respect to the security level specification given in table 5. It can be observed that for 80-bit, 112-bit, 128-bit, 192-bit and 256-bit security level, the corresponding digest sizes of MAC are 160 bits, 224 bits, 256 bits, 384 bits and 512 bits respectively.

In the evaluation of communication overhead, the size of group elements defined in Table 6 can be reduced using the standard compression techniques [30]. Table 4 specifies the corresponding compressed sizes of group elements. Based on this, we obtain the communication overhead (CO) of the schemes SL+BF, LXJ+BF, FL, AJL, CZJZJSZ and DA-ENS, for the given security level, as shown in the following equations:

$$CO_{SL+BF} = 3|\mathbb{G}_1| + |\mathbb{Z}_q^*| + |MAC| + 2|m| \quad (5)$$

$$CO_{LXJ+BF} = 2|\mathbb{G}_1| + |\mathbb{G}_2| + 2|m| \quad (6)$$

$$CO_{FL} = |\mathbb{G}_1| + |\mathbb{G}_2| + |m| \quad (7)$$

$$CO_{AJL} = |\mathbb{G}_1| + |\mathbb{G}_2| + 2|m| \quad (8)$$

$$CO_{CZJZJSZ} = |\mathbb{G}_1| + |\mathbb{G}_2| + |m| \quad (9)$$

$$CO_{DA-ENS} = 2|\mathbb{G}_1| + |MAC| + |m| \quad (10)$$

Table 3 illustrates the communication overhead of all schemes at different security levels. The size of the transmit-

TABLE 7. Comparison of computational cost.

	Sender			*Receiver*		
	$Pairing$	$Mul(\mathbb{G}_1)$	$Exp(\mathbb{G}_2)$	$Pairing$	$Mul(\mathbb{G}_1)$	$Exp(\mathbb{G}_2)$
SL+BF	6	4	3	1	2	0
LXJ+BF	2	4	1	2	1	0
FL	1	3	0	1	1	0
AJL	2	2	0	2	2	0
CZJZJSZ	1	3	0	1	2	0
DA-ENS	0	3	0	0	2	0

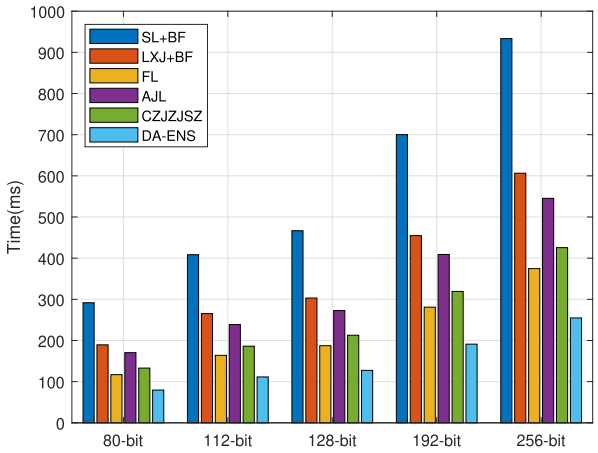

FIGURE 4. Computational time.

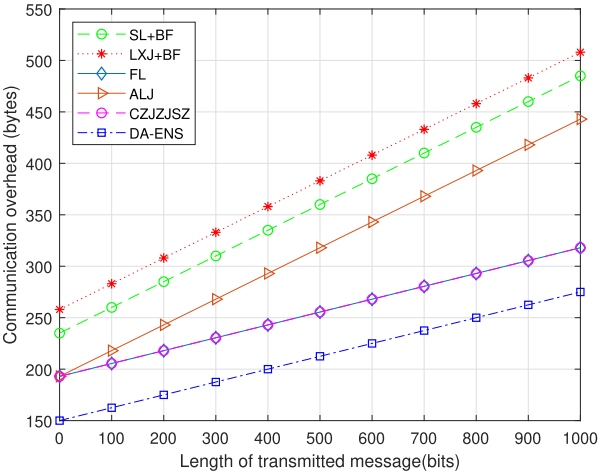

FIGURE 5. 80-bit security level.

ted messages ranges from 0 to 1000 in a step increase of 100. Figure 5 draws the communication overhead of all schemes at the 80-bit security level.

Similarly, figures 6, 7, 8 and 9 show the comparison at 112-bit, 128-bit, 192-bit and 256-bit security levels respectively. We can observe that the **DA-ENS** scheme has relatively less communication overhead than the other schemes.

The evaluation of computational time is based on the total number of cryptographic operations performed in algorithms1 and 2. We consider the computational time of pairing

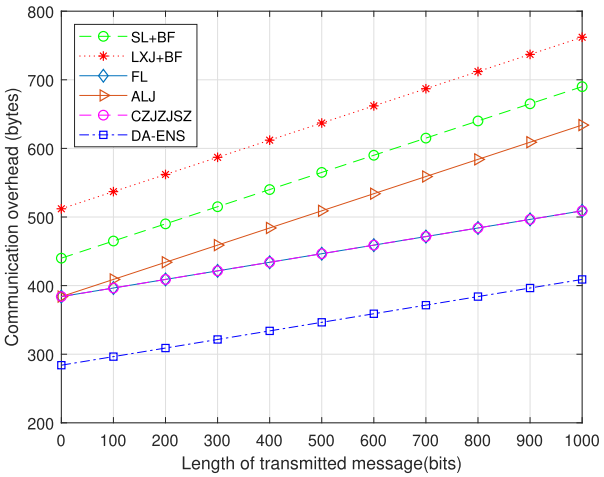

FIGURE 6. 112-bit security level.

TABLE 8. Computational time(ms).

Security level	80-bit	112-bit	128-bit	192-bit	256-bit
SL+BF	291.70	408.38	466.72	700.08	933.4376
LXJ+BF	189.481	265.2734	303.1696	454.7544	606.3372
FL	117.068	163.8952	187.3088	280.9632	374.616
AJL	170.428	238.5992	272.6848	409.0272	545.368
CZJZJSZ	132.995	186.193	212.792	319.188	425.582
DA-ENS	79.635	111.489	127.416	191.124	254.83

computation (PC), point multiplication (PM) in group $\mathbb{G}_1$ and exponent computation (EC) in group $\mathbb{G}_2$. Since the running time of other operations are relatively very low, it is ignored in the evaluation. In addition, we follow the specification given in table 5 and evaluate the computational time for 80-bit, 112-bit, 128-bit, 192-bit and 256-bit security levels.

VII. EXPERIMENTAL ANALYSIS

We implemented our proposed scheme, **DA-ENS**, together with the relevant schemes: SL+BF, LXJ+BF, FL, AJL and CZJZJSZ to compare their performance. We used three performance measures to evaluate and compare the different schemes: computation time, communication overhead, and computational energy. The experiments have been conducted on a Dell Latitude $E6430$ computer with Intel Core i5 3210M, 2.5GHz and 4 GB RAM. We have used PBC library [31] for implementation using Type A pairing constructed on curve

$$y^2 \equiv (x^3 + x) \bmod q$$

TABLE 9. Computational time(ms) out-performance of DA-ENS over other schemes.

Security level	SL+BF	LXJ+BF	AJL	CZJZJSZ	FL
80-bit	0.726996915	0.579720394	0.319754331	0.532735231	0.401218091
112-bit	0.726996915	0.579720394	0.319754331	0.532735231	0.401218091
128-bit	0.726996915	0.579720394	0.319754331	0.532735231	0.401218091
192-bit	0.726996915	0.579720394	0.319754331	0.532735231	0.401218091
256-bit	0.726998355	0.579722306	0.319756764	0.532737528	0.401219976

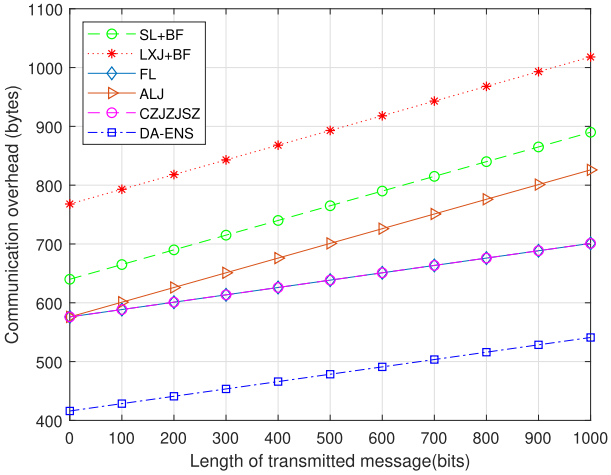

FIGURE 7. 128-bit security level.

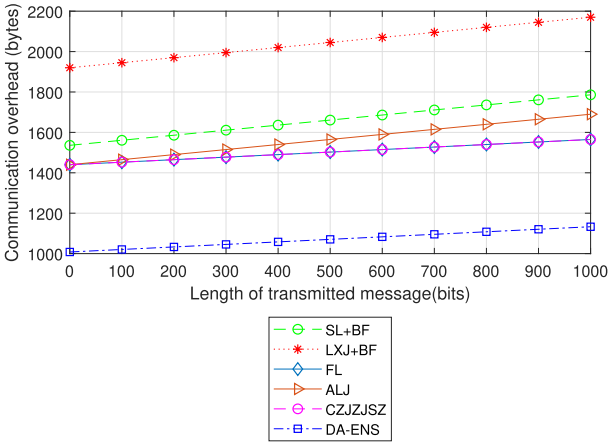

FIGURE 8. 192-bit security level.

for prime $q \equiv 3 \mod 4$, where the embedded degree is 2 and the order of G_1 is p. We follow the implementation process of AES, as in [25], and use the five kind of parameters that represent 80-bit, 112-bit, 128-bit, 196-bit and 256-bit AES key size security levels.

Figure 4 summarizes the running time of these algorithms with respect to the 80-bit, 112-bit, 128-bit, 196-bit and 256-bit security levels. The running time out-performance ratios is computed by using the following equation and is summarized in table 9.

$$O_t = \frac{t_x - t_{DA-ENS}}{t_x} \tag{11}$$

where t_x is the running time of scheme x, and t_{DA-ENS} is the running time of the **DA-ENS** scheme. Given the specifications of the machine were the experiment are conducted, we observed that the execution time of point multiplication in $\mathbb{G}_1$, pairing computation and exponent operation in $\mathbb{G}_2$ require 15.927 ms, 26.68 ms and 3.126 ms respectively. Based on table 7, the running time t_x is computed at all security level, and is summarised in table 8.

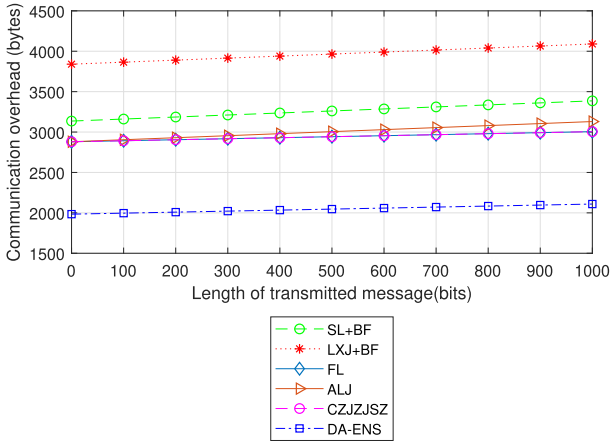

FIGURE 9. 256-bit security level.

After substituting these values in Equation 11. It is found that the **DA-ENS** scheme achieves the least computational times compared to the other schemes at all security levels.

To evaluate the computational energy cost, we adopted the experimental analysis done in [32]–[34]. Let the current drawn in active mode, receiving mode, and transmitting mode be 8.0 mA, 10 mA and 27 mA respectively. MICA2's power level is 3.0 V and data rate is 12.4 kbps. The computational energy cost evaluated in [34], [35] for cryptographic operations is calculated and listed in Table 10.

TABLE 10. Computational energy cost for cryptographic operations in mJ.

Operation	Computational energy cost(mJ)
Pairing	$3:0*8:0*1:9 = 45.6$
Exponent operation in G_2	$3:0*8:0*0:9 = 21.6$
Point multiplication in G_1	$3:0*8:0*0:81 = 19.44$

Then the computational energy at both the sender and the receiver end can be computed as in Table 11.

Out-performance ratios of computational energy cost in percentage is computed by using equation 12

$$O_c = \frac{e_x - e_{DA-ENS}}{e_x} * 100 \tag{12}$$

TABLE 11. Computational energy cost in mJ.

Scheme	Computational energy cost (mJ)		
	Sender	Receiver	Total
DA-ENS	3 * 19.44 = 58.32	2*19.44 = 38.88	97.20
SL+BF	6* 45.60 + 4 * 19.44 + 3 * 21.60 = 416.16	2 *19.44+1 * 45.60+0 * 21.60 = 84.48	500.64
LXJ+BF	4 * 19.44 + 2 * 45:60 + 1 *21.60 = 190.56	1 * 19:44+ 2 * 45.60+0 * 21.60 = 110.60	301.20
FL	3 *19.44+1 * 45.60 = 103.90	1 * 19.44+1 * 45.60+0 * 21.60 = 65.04	168.96
AJL	2 * 19.44+2 * 45.60+0 * 21.60 = 130.08	2*19.44+2*45.60+0*21.60 = 130.08	260.16
CZJZJSZ	3 * 19.44 + 1 * 45.60 + 0 * 21.60 = 103.92	2*19.44+1*45.60+0*21.60 = 84.48	188.40

TABLE 12. Security and communication overhead.

Scheme	Security		Formal Security	Communication overhead
	IND-CCA2	DA-CMA		
SL+BF	No	No	×	$3\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\| + \|MAC\| + 2\|m\|$
LXJ+BF	No	No	×	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 2\|m\|$
FL	Yes	Yes	√	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + \|m\|$
AJL	Yes	Yes	√	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 2\|m\|$
CZJZJSZ	Yes	Yes	√	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + \|m\|$
DA-ENS	Yes	Yes	√	$2\|\mathbb{G}_1\| + \|MAC\| + 2\|m\|$

where e_x is the computational energy cost scheme x, and e_{DA-ENS} is the computational energy cost of the **DA-ENS** scheme. Hence for all schemes SL+BF, LXJ+BF, FL, AJL and CZJZJSZ, the cost O_c in percentage is given by $\frac{500.64-97.20}{500.64} = 80\%$, $\frac{301.20-97.20}{301.20} = 67\%$ $\frac{168.96-97.20}{168.96} = 42\%$ $\frac{260.96-97.20}{260.96} = 62\%$ and $\frac{188.96-97.20}{188.96} = 48\%$ respectively. This is summarized in Table 11 and drawn in Figure 10.

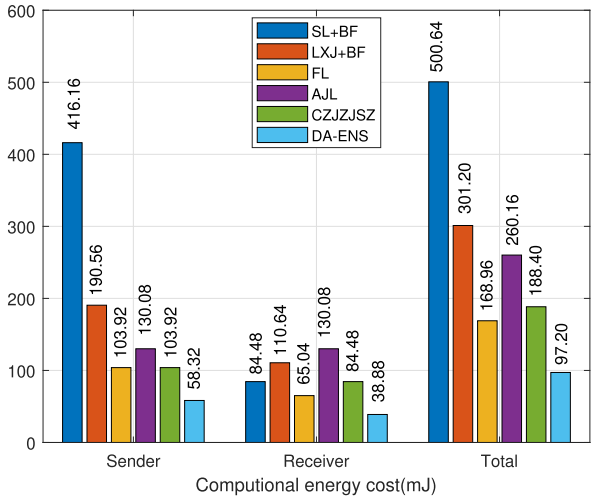

FIGURE 10. Computational energy cost in mJ.

VIII. CONCLUSION

In this paper, we proposed Deniable Authenticated Encryption Scheme (**DA-ENS**), a low-cost protocol with high level of security for e-mail applications. Since **DA-ENS** provides both confidentiality and deniable authentication, we have proven its security against DA-CMA and IND-CCA2 in the random oracle model. We compared the performance of the proposed scheme with the other relevant protocols, taking into consideration the different security levels: 80-bit, 112-bit, 128-bit, 192-bit and 256-bit. We implemented the protocols, and showed that the running time, the communication overhead and the computational energy of **DA-ENS** is relatively less than the other protocols designed for e-mail applications.

APPENDIXES

APPENDIX A

PROOF OF THEOREM 1

Proof: The algorithm $\mathcal{B}$ executes $\mathcal{A}$ as a subroutine and plays $\mathcal{A}$'s challenger in the IND-CCA2 game. $\mathcal{B}$ would take the random instances (P, aP, bP) as input and attempt to compute $W^* = abP$. In fact, this is the contradiction of GDH problem assumption. $\mathcal{A}$ would submit the queries in an adaptive manner to the random oracles of deniably authenticated encryption and decryption. Also $\mathcal{A}$ may consult $\mathcal{B}$ for answer of the queries to random oracles H_1 and H_2. $\mathcal{B}$ constructs two sets as $L_1^{H_1}$ and $L_2^{H_1}$ for H_1 oracle. Similarly constructs $L_1^{H_2}$ and $L_2^{H_2}$ for H_2 oracle. These two sets would store the answers when $\mathcal{A}$ ask by submitting the queries to these oracles. The purpose of this simulation is $\mathcal{B}$ uses these answers from the sets and attempt to compute $W^* = abP$.

`Initial`: At the beginning of the game IND-CCA2, $\mathcal{B}$ initializes the systems parameters *param* by executing **Setup** algorithm. $\mathcal{B}$ selects a random number $k^* \in \{0,1\}^n$ to compute $H(W^*)$, where W^* is unknown to $\mathcal{B}$. Further, $\mathcal{B}$ picks numbers γ^* and λ^* from $\mathbb{Z}_p$ randomly. Constructs sender's public key as $Q_s = \lambda^{*-1}(\gamma^* P - aP)$ and receiver's public key as $b \cdot P$. $\mathcal{B}$ returns Q_s and Q_r and provides to $\mathcal{A}$. There are two possible phases in the simulation.

`Phase-I`: $\mathcal{B}$ answers $\mathcal{A}$'s queries as follows:

H_1 *Queries:* The input and output parameters obtained when, $\mathcal{A}$ submits the queries adaptively to the random oracle H_1 are stored in the list $L_1^{H_1}$. Let the queries submitted to H_1 random oracle is indexed by $i \in \{1 \ldots q_{h_1}\}$. The entries are of the form (W_i^*, k_i^*). Similarly the list $L_2^{H_2}$ stores the parameters $(V_i, ?, k_i)$. The relation between these two input and output parameters can be represented explicitly as $H_1(d_r V_i) = k_i$. Let $d_r V_i$ is denoted by "?", since it is not stored explicitly. For a $H_1(W)$ queries, $\mathcal{B}$ performs the followings:

- If DDH$(P, aP, Q_r, W) = \top$, then terminate and returns W as the solution of GDH problem.
- Else if the oracle DDH$(P, V_i, Q_r, W) = \top$ for some $(V_i, ?, k_i) \in L_1^{H_1}$, then returns k_i
- Else if $W = W_i$ for some (W_i, k_i) in $L_1^{H_1}$, then returns k_i.
- Else picks $k_i \in \{0, 1\}^n$ randomly, add (W, k_i) in $L_1^{H_1}$ and returns k_i

H_2 Queries: $\mathcal{B}$ also constructs two list as $L_1^{H_2}$ and $L_2^{H_2}$ to sores when the queries are being submitted to the random oracle H_2. The input and output entries obtained during this simulation are of the form $(m_i \| Q_s \| Q_r \| W_i, \lambda_i)$. These are stored in $L_1^{H_2}$. Similarly the $L_2^{H_2}$ stores the special type of entries $(V_i, m_i \| Q_s \| Q_r \| ?, \lambda_i)$. The input and output relation is explicitly represented by $H_2(m_i \| Q_s \| Q_r \| d_r V_i) = \lambda_i$. Since $d_r V_i$ is not explicitly stored, it is denoted by "?". For a query $H_2(m \| Q_s \| Q_r \| W)$, $\mathcal{B}$ performs the following steps:

- If the oracle DDH$(P, aP, Q_r, W) = \perp$ then terminate and return W as solution of GDH problem.
- Else if the oracle DDH$(P, V_i, Q_r, W) = \perp$ for some $(V_i, m_i \| Q_s \| Q_r \| ?, \lambda_i)$ in $L_2^{H_2}$, then return λ_i
- Else if $(m \| Q_s \| Q_r \| W, \lambda_i)$ is in $L_1^{H_2}$ and returns λ_i.
- Else picks $\lambda_i \in \mathbb{Z}_p^*$ randomly and add $(m \| Q_s \| Q_r \| W, \lambda_i)$ in $L_1^{H_2}$ and returns λ_i.

Deniably Authenticated Encryption Queries: In the simulation, when $\mathcal{A}$ submits the queries on message m to deniably authenticated encryption random oracle, $\mathcal{B}$ performs the following steps:

- Picks a number $k \in \{0, 1\}^n$ randomly and computes $c = m \oplus k$.
- $\mathcal{B}$ selects $\lambda, \gamma \in \mathbb{Z}_q^*$ as input and computes $V = \lambda P - \gamma Q_s$ and add the entry $(V, "?", k)$ to $L_2^{H_1}$ and $(m \| Q_s \| Q_r \| ?, \lambda)$ to $L_2^{H_2}$.
- At the end $\mathcal{B}$ computes $U = \gamma P$ and $R = \gamma Q_r$ and sends the ciphertext $\sigma = (c, \lambda, U, R)$ to $\mathcal{A}$.

Deniably Authenticated Decryption Queries: $\mathcal{A}$ submits the queries on the ciphertext $\sigma == (c, \lambda, U, R)$ in adaptive manner then $\mathcal{B}$ performs the followings:

- Computes $V = U - \lambda Q_r$
- If $V = aP$, then abort.
- If there exists (W_i, k_i) in $L_1^{H_1}$ such that the oracle DDH$(P, V, Q_r, W_i) = \perp$ or $(V_i, ?, k_i)$ in $L_2^{H_1}$ such that $V = V_i$, set $k' = k_i$.
- Else picks k' from $\{0, 1\}^n$ randomly and add $(V, ?, k')$ into $L_2^{H_1}$.
- Computes $m = c \oplus k'$.
- if there exists $(m_i \| Q_s \| Q_r \| W_i, \lambda_i)$ in $L_1^{H_2}$ such that DDH$(P, V, Q_r, W_i) = \perp$ or otherwise there exists $(V_i, m_i \| Q_s \| Q_r \| ?, \lambda_i)$ in $L_2^{H_2}$ such that $V = V_i$ and $m = m'$ for some λ_i set $\lambda' = \lambda_i$.
- Else picks randomly $\lambda' \in \mathbb{Z}_q^*$ and add $(V, m \| Q_s \| Q_r \| ?, \lambda')$ in $L_2^{H_2}$.
- If $\lambda = \lambda'$ and DDH$(P, U, Q_r, R) = \perp$ then return m.
- Else abort.

`Challenge`: $\mathcal{A}$ picks two plain text m_0 and m_1. $\mathcal{B}$ picks a random bits $\xi \in \{0, 1\}$ and encrypts m_ξ. $\mathcal{A}$ performs the following steps to obtain the ciphertext.

- Compute $c^* = m_\xi \oplus k^*$
- Compute $U^* = \gamma^* P$ and $R^* = \gamma^* Q_r$

Return the ciphertext $\sigma^* = (c^*, \lambda^*, U^*, R^*)$.

`Phase-II`: $\mathcal{A}$ submits second series of queries to all the oracles except deniably authenticated decryption oracle. on the challenged ciphertext σ^* obtained.

`Guess`: At the end of the IND-CCA2 game, $\mathcal{A}$ produces a bit ξ' as it guess. Then $\mathcal{B}$ returns W^* which is a guess for abP and is a pre-image of k^*.

`Probability of Success`

Here we do the analysis of $\mathcal{B}$'s success. Consider the following events:

- let E_0 be the event occurs when $\mathcal{A}$ asks $H_1(W^*)$ during simulation. This would be same as real attack if the attack environment continues ideally. In real attack, we have

$$\begin{aligned} Pr[\xi = \xi'] &\leq Pr[\xi = \xi' | \neg E_0] Pr[\neg E_0] + Pr[E_0] \\ &= \frac{1}{2}(1 - Pr[E_0]) + Pr[E_0] \\ &= \frac{1}{2} + \frac{1}{2} Pr[E_0]. \end{aligned}$$

  Therefore, $2\,\Pr[\xi = \xi'] - 1 \leq \Pr[E_0]$ Further, we note that the simulation only gets fails to provide a consistent simulation because on of the occurrence of following independent events.
- E_1: $\mathcal{B}$ aborts in a deniably authenticated encryption query due to the collision on H_1 and H_2.
- $\mathcal{B}$ rejects a valid ciphertext at some instant during the game in deniably authenticated decryption query.

We have $\Pr[\neg E_1] \leq q_e \frac{(q_{h_1} + q_{h_2})}{2^\theta}$. where θ is considered as security parameter such that both h_1 and h_2 are uniformly taken from a set of 2^θ elements.

$\Pr[E_2] \leq \frac{q_d}{2^\theta}$. Therefore,

$$\begin{aligned} \Pr[\neg E_1] + \Pr[E_2] &= \frac{q_e(q_{h_1} + q_{h_2}) + q_d}{2^\theta} \\ \implies \epsilon_{dae} &\leq \epsilon_{gdh} + \frac{q_e(q_{h_1} + q_{h_2}) + q_d}{2^\theta} \end{aligned}$$

□

APPENDIX B
PROOF OF THEOREM 2

Proof: Let $\mathcal{B}$ takes the random instances (a, aP, bP) as input and tries to compute abP, In DA-CMA game, $\mathcal{B}$ uses $\mathcal{F}$ as subroutine and plays $\mathcal{B}$'s challenger. $\mathcal{B}$ submits the series of queries to random oracles H_1, H_2, deniably authenticated encryption and decryption oracles in an adaptive manner. In the simulation, $\mathcal{B}$ constructs two lists as $L_1^{H_1}$ and $L_1^{H_1}$ to store both the input and output of the queries submitting to the random oracles H_1 and H_2 respectively.

`Initial`: At the beginning of DA-CMA game, $\mathcal{B}$ executes the **Setup** algorithm and obtains the systems

parameters *param*. Then $\mathcal{B}$ constructs sender's public key $Q_s = d_sP$ and receiver's public key $Q_r = d_rP$. $\mathcal{B}$ provides these two parameters Q_s and Q_r to $\mathcal{F}$.

`Attack`: In the DA-CMA game, the series of queries submitted in an adaptive manner to the random oracles H_1, H_2, deniably authenticated encryption and decryption are described bellows:

H_1-Queries: $\mathcal{B}$ maintains the as $L_1^{H_1}$ to store the element of the form (W_i, ki_i) that obtains when $\mathcal{F}$ asks the queries to the oracle H_1 $L_1^{H_2}$. Similarly an another list $L_2^{H_1}$ is maintained by $\mathcal{B}$ to store the input/out entries of the form $(V_i, ?, k_i)$ obtained when $\mathcal{F}$ submits the queries to the oracle H_2. The input/output is implicitly represented by the relation $H_1(d_rV_i) = k_i$. Le d_rV_i is denoted by "?", where the index $i \in \{1 \dots q_{h_1}\}$. Since is was not stored explicitly. For $H_1(W)$ query, $\mathcal{B}$ performs the following:

- If DDH$(P, V_i, Q_r, W) = \top$ for some $(V_i, ?, k_i) \in L_2^{H_1}$, then return k_i.
- Else if $W = W_i$ for some (W_i, k_i) in $L_1^{H_1}$ then return k_i.
- Else picks randomly $k_i \in \{0, 1\}^n$, add (W, k_i) into $L_1^{H_1}$ and return k_i.

H_2-Queries: Similarly, $\mathcal{B}$ stores all entries of input and output in the list $L_1^{H_2}$. These are obtained when the queries are submitted to random oracle H_2. The elements of the list $L_1^{H_2}$ are of the form $(m_i\|Q_s\|Q_r\|W_i, \lambda_i)$ and other special kind of entries are of the form $(V_i, m_i\|Q_s\|Q_r\|?, \lambda_i)$. The relation between the input and output is represented as $H_2(m_i\|Q_s\|Q_r\|d_rV_i) = \lambda_i$, where d_rV_i is denoted by "?". Since it is not stored explicitly. For a query $H_2(m\|Q_s\|Q_r\|W)$, $\mathcal{B}$ performs the following:

- If DDH$(P, V_i, Q_r, W) = \top$ for some $(V_i, m_i\|Q_s\|Q_r\|W, \lambda_i)$ is in $L_1^{H_2}$, return λ_i.
- Else if $(m\|Q_s\|Q_r\|W, \lambda_i)$ is in $L_1^{H_2}$ return λ_i.
- Else chooses randomly $\lambda_i \in \mathbb{Z}_q^*$, add $(m\|Q_s\|Q_r\|W, \lambda_i)$ into $L_1^{H_2}$ and return λ_i.

Deniably Authenticated Encryption Queries: When $\mathcal{F}$ submits queries to deniably authenticated encryption oracle on the message m, $\mathcal{B}$ picks a random $k \in \{0, 1\}^n$ and computes the ciphertext $c = m \oplus k$. Then $\mathcal{B}$ picks randomly $\lambda, \gamma \in \mathbb{Z}_q^*$ and computes $V = \gamma P - \lambda Q$.. $\mathcal{B}$ adds (V, P, k) into $L_2^{H_1}$ and $(m\|Q_s\|Q_r\|?, \lambda)$ into $L_2^{H_2}$. Finally $\mathcal{B}$ computes $U = \gamma P$ and $R = \gamma Q_r$. Then sends the ciphertext $\sigma = (c, \lambda, U, R)$ to $\mathcal{F}$.

Deniably Authenticated Decryption Queries: $\mathcal{F}$ submits a series od queries to deniably authenticated decryption oracle in an adaptive manner on the ciphertext $\sigma = (c, \lambda, U, R)$. $\mathcal{B}$ performs the following:

- Computes $V = U - \lambda Q_s$
- If there exists an entry (W_i, k_i) in $L_1^{H_1}$ such that DDH$(P, V, Q_r, W_i) = \top$ or $(V_i, ?, k_i)$ in $L_2^{H_1}$ such that $V = V_i$, set $k' = k_i$.
- Else picks randomly $k' \in \{0, 1\}^n$ add $(V, ?, k')$ into $L_2^{H_1}$.
- Compute $m \oplus k'$.
- If the entry $(m_i\|Q_s\|Q_r\|W_i, \lambda_i)$ exists in $L_1^{H_2}$ such that DDH$(P, V, Q_r, W) = \top$ or $\exists(V_i, m_i\|Q_s\|Q_r\|?, \lambda_i)$ in $L_2^{H_2}$ such that $V = V_i$ and $m = m_i$ for some λ_i, set $\lambda' = \lambda_i$.
- Else $(V, m\|Q_s\|Q_r\|?, \lambda')$ in $L_2^{H_2}$.
- $\lambda = \lambda'$ and DDH$(P, U, Q_r, R) = \top$ then return m.
- Else abort the simulation.

`Forgery`: Finally $\mathcal{F}$ obtains a ciphertext $\sigma' = (c', \lambda', U', R')$. During the simulation, it is required to check whether the hash value $H_2(m\|Q_s\|Q_r\|W')$ has been queried or not. If not, then $\mathcal{B}$ fails and terminate. Otherwise, $\mathcal{B}$ searches the entry (W', λ') in $L_1^{H_2}$ and $L_2^{H_2}$. Then $\mathcal{B}$ solves DDH problem by computing $-\lambda'^{-1}(W' - R')$. Since $W' = u'Q_r$, $R' = \gamma'Q_r$ and $\gamma' = (u' + \lambda' d_s) \bmod q$. We have

$$\begin{aligned} -\lambda'^{-1}(W' - R') &= -\lambda'^{-1}(u'Q_r - \gamma'Q_r) \\ &= -\lambda'^{-1}(u'd_rP - \gamma'd_rP) \\ &= -\lambda'^{-1}(u' - \gamma')d_rP = d_sQ_r = abP \end{aligned}$$

Probability of Success

Consider the following events to compute $\mathcal{B}$'s probability of success.

- Let E_0 be the event that $\mathcal{F}$ wins the game obtaining a forge ciphertext $\sigma' = (c', \lambda', U', R')$ without asking the query $H_2(m'\|Q_s\|Q_r\|W)$. Hence $\Pr[E_0] \leq \frac{1}{2^\theta}$. This would be failed in providing a consistent simulation because one of the following events occur.
- E_1: $\mathcal{B}$ terminates the query submitting to denaibly authenticated encryption oracle because of collision on H_1 and H_2.
- E_2: $\mathcal{B}$ rejects a valid ciphertext in a query to deniably authenticated decryption oracle.

Therefore $\Pr[E_1] = q_e \frac{q_{h_1} + q_{h_2}}{2^\theta}$ and $\Pr[E_2] = \frac{q_d}{2^\theta}$.

Hence

$$\begin{aligned} &\Pr[E_0] + \Pr[E_1] + \Pr[E_2] \\ &\quad = \frac{1}{2^\theta} + q_e\frac{q_{h_1} + +q_{h_2}}{2^\theta} + \frac{q_d}{2^\theta} \\ &\quad = \frac{q_e(q_{h_1} + q_{h_2}) + q_d + 1}{2^\theta} \\ &\quad \Longrightarrow \epsilon_{dae} \leq \epsilon_{gdh} + \frac{q_e(q_{h_1} + q_{h_2}) + q_d + 1}{2^\theta} \end{aligned}$$

□

REFERENCES

[1] A. Nenadić, N. Zhang, and S. Barton, "Fair certified e-mail delivery," in *Proc. ACM Symp. Appl. Comput.*, 2004, pp. 391–396.
[2] L. Harn and J. Ren, "Design of fully deniable authentication service for e-mail applications," *IEEE Commun. Lett.*, vol. 12, no. 3, pp. 219–221, Mar. 2008.
[3] J. Kar, "ID-based deniable authentication protocol based on Diffie–Hellman problem on elliptic curve," *IJ Netw. Secur.*, vol. 15, no. 5, pp. 357–364, 2013.
[4] J. Klensin, *Simple Mail Transfer Protocol*, document RFC 5321, Oct. 2008.
[5] B. Ramsdell, *Secure/Multipurpose Internet Mail Extensions (S/MIME)*, document RFC 5751, Jan. 2010.
[6] D. Shaw, *The Camellia Cipher in Openpgp*, document RFC 4880, Jun. 2009.
[7] D. W. Chadwick, A. J. Young, and N. K. Cicovic, "Merging and extending the PGP and PEM trust models-the ICE-TEL trust model," *IEEE Netw.*, vol. 11, no. 3, pp. 16–24, May/Jun. 1997.
[8] S. L. Garfinkel, D. Margrave, J. I. Schiller, E. Nordlander, and R. C. Miller, "How to make secure email easier to use," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2005, pp. 701–710.

[9] F. Li, Z. Zheng, and C. Jin, "Identity-based deniable authenticated encryption and its application to e-mail system," *Telecommun. Syst.*, vol. 62, no. 4, pp. 625–639, Aug. 2016.
[10] F. Li, P. Xiong, and C. Jin, "Identity-based deniable authentication for ad hoc networks," *Computing*, vol. 96, no. 9, pp. 843–853, 2014.
[11] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *J. Cryptol.*, vol. 21, no. 4, pp. 469–491, Oct. 2008.
[12] H. Petersen, "Authenticated encryption schemes with low communication costs," *Electron. Lett.*, vol. 30, no. 15, pp. 1212–1213, Jul. 1994.
[13] Y. Aumann and M. O. Rabin, "Authentication, enhanced security and error correcting codes," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 1998, pp. 299–303.
[14] S. Zeng, S. Tan, Y. Chen, M. He, M. Xia, and X. Li, "Privacy-preserving location-based service based on deniable authentication," in *Proc. IEEE/ACM 9th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2016, pp. 276–281.
[15] J. Kar and B. Majhi, "A novel deniable authentication protocol based on Diffie–Hellman algorithm using pairing technique," in *Proc. Int. Conf. Commun., Comput. Secur.*, 2011, pp. 493–498.
[16] B. Ramsdell and S. Turner, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*, document RFC 3851, 2004.
[17] S. Garfinkel, *PGP: Pretty Good Privacy*. Newton, MA, USA: O'Reilly Media, 1995.
[18] M. F. Zibran, "Cryptographic security for emails: A focus on S/MIME," Dept. Comput. Sci., Univ. Saskatchewan, Saskatoon, SK, Canada, Tech. Rep. 2011-03, 2011.
[19] D. Adam, "Enterprise smart card deployment in the microsoft windows smart card framework," Microsoft, Redmond, WA, USA, Tech. Rep., vol. 26, Jun. 2006.
[20] C. Meyer, J. Somorovsky, E. Weiss, J. Schwenk, S. Schinzel, and E. Tews, "Revisiting SSL/TLS implementations: New bleichenbacher side channels and attacks," in *Proc. USENIX Secur. Symp.*, 2014, pp. 733–748.
[21] J. Ki, J. Y. Hwang, D. Nyang, D. H. Lee, and J. Lim, "Privacy-enhanced deniable authentication e-mail service," in *Digital Enterprise and Information Systems*. Springer, 2011, pp. 16–29.
[22] L. Harn, C.-Y. Lee, C. Lin, and C.-C. Chang, "Fully deniable message authentication protocols preserving confidentiality," *Comput. J.*, vol. 54, no. 10, pp. 1688–1699, 2011.
[23] J. Kar, "Non-interactive deniable authentication protocol using generalized ECDSA signature scheme," in *Proc. Int. Conf. Inf. Secur. Assurance*. Springer, 2011, pp. 166–176.
[24] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, Jan. 2004.
[25] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer, 2013.
[26] Y. Shi and J. Li, "Identity-based deniable authentication protocol," *Electron. Lett.*, vol. 41, no. 5, pp. 241–242, Mar. 2005.
[27] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 2001, pp. 213–229.
[28] E. Ahene, C. Jin, and F. Li, "Certificateless deniably authenticated encryption and its application to E-voting system," *Telecommun. Syst.*, vol. 70, no. 3, pp. 417–434, Mar. 2019.
[29] G. Chen, J. Zhao, Y. Jin, Q. Zhu, C. Jin, J. Shan, and H. Zong, "Certificateless deniable authenticated encryption for location-based privacy protection," *IEEE Access*, vol. 7, pp. 101704–101717, 2019.
[30] K.-A. Shim, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1874–1883, May 2012.
[31] B. Lynn. (2006). *PBC Library*. [Online]. Available: http://crypto.stanford.edu/pbc
[32] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2004, pp. 119–132.
[33] K.-A. Shim, Y.-R. Lee, and C.-M. Park, "EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 182–189, Jan. 2013.
[34] C. Ma, K. Xue, and P. Hong, "Distributed access control with adaptive privacy preserving property for wireless sensor networks," *Secur. Commun. Netw.*, vol. 7, no. 4, pp. 759–773, Apr. 2014.
[35] K.-A. Shim, "S2DRP: Secure implementations of distributed reprogramming protocol for wireless sensor networks," *Ad Hoc Netw.*, vol. 19, pp. 1–8, 2014.

JAYAPRAKASH KAR received the M.Sc. and M.Phil. degrees in mathematics from Sambalpur University and the M.Tech. and Ph.D. degrees in computer science (cryptographic protocols) from Utkal University, India. He is currently a Visiting Researcher with the Department of Electrical & Computer Engineering, University of Waterloo, Canada. He is also an Associate Professor with the Department of Computer Science & Engineering, The LNM Institute of Information Technology, Jaipur, India. He is co-lead of Center for Cryptography, Cyber Security and Digital forensics, LNMIIT, Jaipur, India. His current research interests are on Cryptographic protocols and primitives using Elliptic Curve and Pairing based Cryptography in Random Oracle and Standard model. Dr. Kar is a Life Member of International Association for Cryptology Research (IACR), Cryptology Research Society of India, International Association of Computer Science & Information Technology, Singapore, International Association of Engineers, USA, and a Professional Member of IEEE and ACM. He has served as a Program Chair of many international conferences. He is an Associate Editor of the *Journal of Circuits, Systems, and Computers* and an Editorial Board Member of many peer-reviewed Journals.

KSHIRASAGAR NAIK held faculty positions at Carleton University, Ottawa, and the University of Aizu, Japan. He worked as a Software Developer for three years in Wipro, Bangalore - one of the largest software consultancy companies in the world. He is currently a Full Professor with the Department of Electrical and Computer Engineering, University of Waterloo. He is a coauthor of two widely used textbooks, namely, *Software Testing and Quality Assurance: Theory and Practice* (Wiley, 2008) and *Software Evolution and Maintenance: A Practitioner's Approach* (Wiley, 2014). His research interests include vehicular networks, delay tolerant networks, energy performance testing of mobile applications, detection of anomalous behavior of wireless devices and physical systems, the energy harvesting Internet of Things (IoT) devices for sustainable monitoring of physical systems, communication security, and communication protocols for smart power grids. Designing mathematical models and building prototype sensor networks for performing real-life, controlled experiments lie at the core of his research. He has served on the editorial boards of many journals, including the *Journal of Peer-to-Peer Networking and Applications*, the *International Journal of Parallel, Emergent and Distributed Systems*, the *Journal of Circuits, Systems, and Computers*, and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He was a Co-Guest Editor of four special issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the IEEE TRANSACTIONS ON CLOUD COMPUTING.

TAMER ABDELKADER received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree in computer and information sciences from Ain Shams University, Cairo, Egypt, in 2003, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Ontario, Canada, in 2012. After graduation, he was with the University of Waterloo as a Postdoctoral Researcher, and a Visiting Researcher. He was the Manager of the Information and Technology Research Consultancy Center (ITRCC), Ain Shams University, Cairo, Egypt. He was an Information and Technological Consultant in several governmental and private companies, including the Information and Communication Technology Project (ICTP), Egypt, and the Ministry of Electricity. He is currently an Associate Professor and the Vice Dean for community services and environmental affairs in the faculty of Computer and Information Sciences, Ain Shams University. He is the author of several publications in IEEE and other ranked journals and conferences. His current research interests include network and information security, delay tolerant networks, resource allocation in wireless networks, vehicular networks, and energy efficient protocols.