Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 5, No. 1, February, 2021

Improved Identity Based Encryption System (IIBES): A Mechanism for Eliminating the Key-Escrow Problem

Maitri Patel [1, 2*], Rajan Patel [2]

[1] *Faculty of Engineering & Technology, Sankalchand Patel University, Gujarat, India*

[2] *Department of Computer Engineering, Gandhinagar Institute of Technology, Gujarat, India*

Abstract

A revolutionary change to public-key cryptography can be considered as an Identity Based Cryptography (IBC) in which identity of the receiver is being used as a public key for encrypting a message and Key Generation Centre (KGC). IBC will generate and distribute the private key to each user to decrypt a message. The thought behind presenting the scheme was to improve and reduce the complexity of certificate and key management, but it also gives rise to key escrow problem, access to encrypted information to unauthorized users. The paper represents Improved Identity-Based Encryption Scheme (IIBES) for Domain Name System (DNS) security which provides confidentiality and authentication through modified identity based encryption and identity based digital signatures. The IIBES comprises key revocation mechanism for non-revoked users and also eliminates key escrow problem. Thus, the IIBES aids to implement the identity-based cryptography more safely in reality and protects DNS against cache poisoning, spoofing attack and masquerade attack.

Keywords:

Key Escrow Problem;

Identity Based Cryptography;

Key Generation Centre;

Key Revocation; Domain Name System.

Article History:

Received:	15	November	2020
Revised:	19	December	2020
Accepted:	12	January	2021
Published:	01	February	2021

1- Introduction

Now a days with the increasing use of Internet, the Domain Name System (DNS) has become the crucial part of Internet [1] and there is a need to secure DNS as server vulnerabilities exist in today's world [2]. DNS provides mapping of IP addresses (random, hard-to-remember numbers) to host names (easier to remember and disseminate) [1, 3]. But, DNS doesn't include any security mechanism to provide confidentiality and authentication to the data transferred through DNS transaction [3]. DNSSEC protocol is designed to provide authenticity and integrity to the data transferred by DNS through the use of public key encryption algorithms, but it is not fully deployed [4].

Shamir introduced an idea of Identity (ID) based on public key system to get rid of the disadvantage of certificate management for traditional public-key cryptography in which certificates are used to map the identities and public keys to perform the encryption process [5-9]. User's identity like E-mail address, social security number, or name is being used as the public key of user in an ID-based public key system [5, 8-10]. An Identity Based Encryption (IBE) scheme uses key authority, sender, and receiver to accomplish encryption and decryption process [11]. To encrypt and decrypt a message, the sender makes use of the receiver's identity id as the public key and the receiver uses his secret key skid respectively [11]. In an IBE scheme, sender makes use of receiver's identity as a public key [12] which simplifies the key management process as it does not require to ask for the authenticated public key from key authority [11].

[*] **CONTACT**: Maitru1487288@gmail.com

DOI: http://dx.doi.org/10.28991/esj-2021-01259

An IBE scheme simplifies the key management process, as the sender uses the receiver's identity as a public key [12], there is no need to ask for the authenticated public key from key authority [11].

In the BF-IBE system, there are two main roles i.e. Private Key Generator (PKG) and user that all are on the same level. Users will authenticate themselves to PKG to get their corresponding private keys which are generated by PKG [5, 13]. Though IBE provides a great advantage over traditional Public Key Infrastructure (PKI), an efficient key revocation mechanism will be needed to perform key revocation process periodically [14, 15]. When the private key of the user gets compromised, one more problem of revoking the user exists in an existing ID-based encryption system [7, 14, 16]. BF has also tried to solve the issue by suggesting that users should periodically renew their private key through KGC whether the private key of the user has been compromised or not [16]. The suggested solution has the disadvantage of increasing load on KGC with the increase in the number of users as the scheme contains only one KGC for generating the private key of the user [16].

After this, a numerous literature has been published on ID-based cryptography to overcome the disadvantage of BF-IBE System [5]. Boldyreva et al. (2008) [17] introduced an IBE scheme in which KGC only generates partial public-private key pair. From that partial public-private key pair, users will generate their public-private key pair by use of common secret which is known by them. This will eliminate the full dependency on KGC for generating public-private key pair [16]. The scheme tries to eliminate the inherent key escrow problem [18-20] and also aims to generate revocable public-private key pairs [16] Mediators [21], a semi-trusted third party, have also been used to address this problem. In this, mediators will help users in the decryption process by having a partial share of private keys in all users. But, the solution is not adopted widely as in the proposed solution, mediators stop working if the identity of the user gets revoked [16].

In this paper, we tried to provide security to DNS by use of IBE and also tried to address the solution for key revocation and key escrow problem in existing IBE by the use of shared secret and sub KGC. The structure of the remaining paper is as per following: Theoretical background associated to IIBES is explained in section 2. Section 3 provides detailed description and working of IIBES. In section 4, results and discussion for IIBES is presented. Conclusion is presented in Section 5.

2- Theoretical Background

2-1- Algorithm to Select sKGC

To select sKGC modified KUNode algorithm [22] will be used. In that, 'selected' parameter is initially set to false for all sKGC. Then a random number has been generated and sKGC from 1 to n is being selected. Also, 'selected' parameter is set to true (Only for time duration T i.e. single session). After that, sKGC is being revoked by using a modified KUNode algorithm. i.e. set the parameter selected to false. This process will get repeated to select sKGC.

2-2- Modified KUNode Algorithm

Initially, sKGCs will have the value false for the parameter selected and are allocated as leaf node η to a complete binary tree. A period T, the user revokes sKGC for a set KUNode(BT, SL, T).

A binary tree /bt, sKGC list SL, and time T are given as input to the algorithm and a set of nodes will be produced as output. The algorithm can be described as follows: Left and right child of a non-leaf node η are denoted by η_{left} and η_{right} respectively. All sKGCs are allocated as the leaf node. If an sKGC (assigned to η) is revoked on time T, at that point (η, T) ∈ SL. Path(η) indicates the set of nodes on the path from η to root. KUNode can be described as follows:

```
KUNode(BT,$SL$, $T$):
X, Y $\leftarrow \emptyset$;
$\forall(\eta_i, T_i) \in SL$
If $T_i >= T$ and selected=true then
    add Path($\eta_i$) to X
    set Selected($\eta_i$) to false
$\forall_x \in$ X
If $x_{left} \notin$ X then add $x_{left}$ to Y
If $x_{right} \notin$ X then add $x_{right}$ to Y
If Y = $\emptyset$ then add root to Y
Return Y
```

To understand KUNode(BT,SL,T), refer the below figure. Let a sub KGC u3 (assigned to x10) be revoked as shown in example. Then, X = Path(x10) = {x10, x5, x2, root = x1}, and Y ={x3, x4, x11}. Intuitively, all sub KGC except u3, have a node x ∈ Y that is contained in the set of nodes on the path from their assigned node to root: e.g., x4 for u1 and u2, x11 for u4, and x3 for u5, u6, u7, and u8, whereas Y ∩ Path(x10) = ∅.

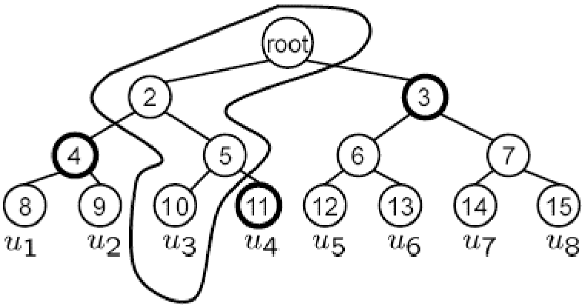


Figure 1. **KUNode Algorithm Example [22].**

2-3- Algorithm to Deliver Shared Secret

Following Shamir's secret sharing algorithm [23] will be used to deliver shared secret in IIBES:

```
START
n ←positive integer (group size), t ← positive integer (threshold value)
p ← bigint()                    /*prime number*/
m ←positive integer             /*minimum number for the secret ( typically > n )*/
s ←bigint                       /*random number between m and p ( m > s < p )*/
shares [n] [2] ←bigint     /*a 2d array to store the calculated shares*/
tshares [t] [2] ←bigint    /*a 2d array to store t shares for reconstruction*/

/* Functions */
SECRET-SPLIT (n, t, s, p)
SECRET-RECONSTRUCT (tshares [], p, t)

SECRET-SPLIT (n, t, s, p)
for i← 0 to t-1 do
        coeff [i] ←random number from 1 to p
end for
for x ← 1 to n do
        share ←s
        for expo 0← to t-1 do
        share ← (share+ (coeff [expo]*(xexpo %p)) %p) %p
        end for
        shares [x-1][0] = x, shares[x-1][1] = share
end for
return shares

SECRET-RECONSTRUCT ( tshares [], p, t)
if length(tshares []) < t then
        print reconstruction not possible
else
        for i ←0 to length (tshares []) do
        xarray [i] ←tshares[i][0]
        yarray [i] ←tshares [i][1]
        end for
        LAGRANGE-INTERPOLATION (xarray [], yarray [], p)
        print secret
end if
LAGRANGE-INTERPOLATION (xarray [], yarray [], p)
secret ←0, sum ←0
for i ←0 to length (yarray []) do
        product ←yarray [i]
        for j ←0 to length (yarray []) do
        if i != j then
```

```
            product ← product * (x xarray [i]) / (xarray [i] - xarray [j])
            end if
            end for
            sum ← sum + product
    end for
    secret ← sum
    return secret
    END
```

3- Improved Identity Based Encryption System without Key-Escrow Problem (IIBES)

Figure 2 shows the system flow of proposed system (IIBES). The Send_Query procedure generates DNS query ID and encrypts the ID using the public key of local server. Also, it brings the NS from the URL and transmits Query (NS and encrypted ID) to DNS server. The Receive_Query procedure searches the database for the NS when Query (ID and NS)arrives.; if found, it fetches the IP and decrypts the ID using the private key of local server. The Send_Response procedure encrypts the IP using the public key of client and transmits Response (ID, NS, and encrypted IP). The Receive_Response procedure compares the received ID with the stored ID when response (ID, NS and encrypted IP) arrives; if they are the same, it will decrypt the IP using the private key of client.

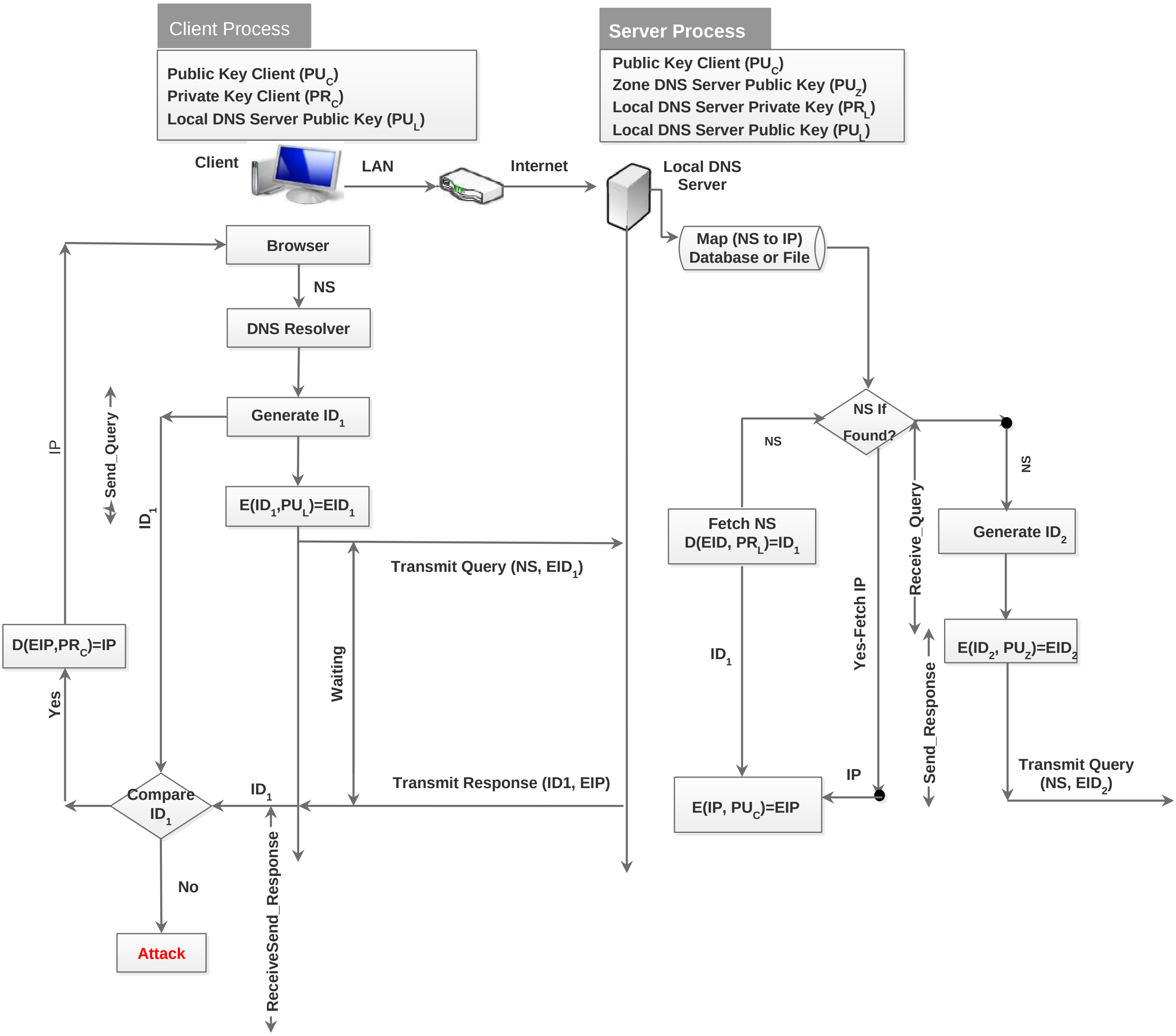

Client Process
Public Key Client (PU_C)
Private Key Client (PR_C)
Local DNS Server Public Key (PU_L)
Server Process
Public Key Client (PU_C)
Zone DNS Server Public Key (PU_Z)
Local DNS Server Private Key (PR_L)
Local DNS Server Public Key (PU_L)
Client
LAN
Internet
Local DNS Server
Map (NS to IP) Database or File
Browser
NS
DNS Resolver
Generate ID_1
$E(ID_1,PU_L)=EID_1$
Transmit Query (NS, EID_1)
IP
Send_Query
ID_1
$D(EIP,PR_C)=IP$
Yes
Waiting
Compare ID_1
ID_1
Transmit Response (ID1, EIP)
No
Attack
ReceiveSend_Response
NS If Found?
NS
Fetch NS $D(EID, PR_L)=ID_1$
ID_1
Yes-Fetch IP
Receive_Query
NS
Generate ID_2
$E(ID_2, PU_Z)=EID_2$
Send_Response
IP
$E(IP, PU_C)=EIP$
Transmit Query (NS, EID_2)

Figure 2. System Flow of Proposed System (IIBES).

This paper tries to improve and eliminate some flaws from the BF-IBE scheme. The solution to the key escrow problem has been addressed by introducing a set of subkey generation centers (sKGC), a trusted third party. The sender can choose any of the existing sKGCs by using the modified KUNode algorithm which has been explained in section 2[22]. sKGC is responsible to generate a shared secret (random number and offset) which will be delivered to the sender and receiver by using Shamir's secret sharing algorithm as explained in section 2[23,24]. The receiver's location will be provided to sKGC by the sender itself. To accomplish encryption and decryption process, both sender and receiver uses shared secret. To resolve key revocation problem, nonce has been used in shared secret. The user has to revoke his/her shared secret after the nonce gets expired. The IIBES also makes use of the Adi Shamir's digital signature algorithm to provide authenticity [25].

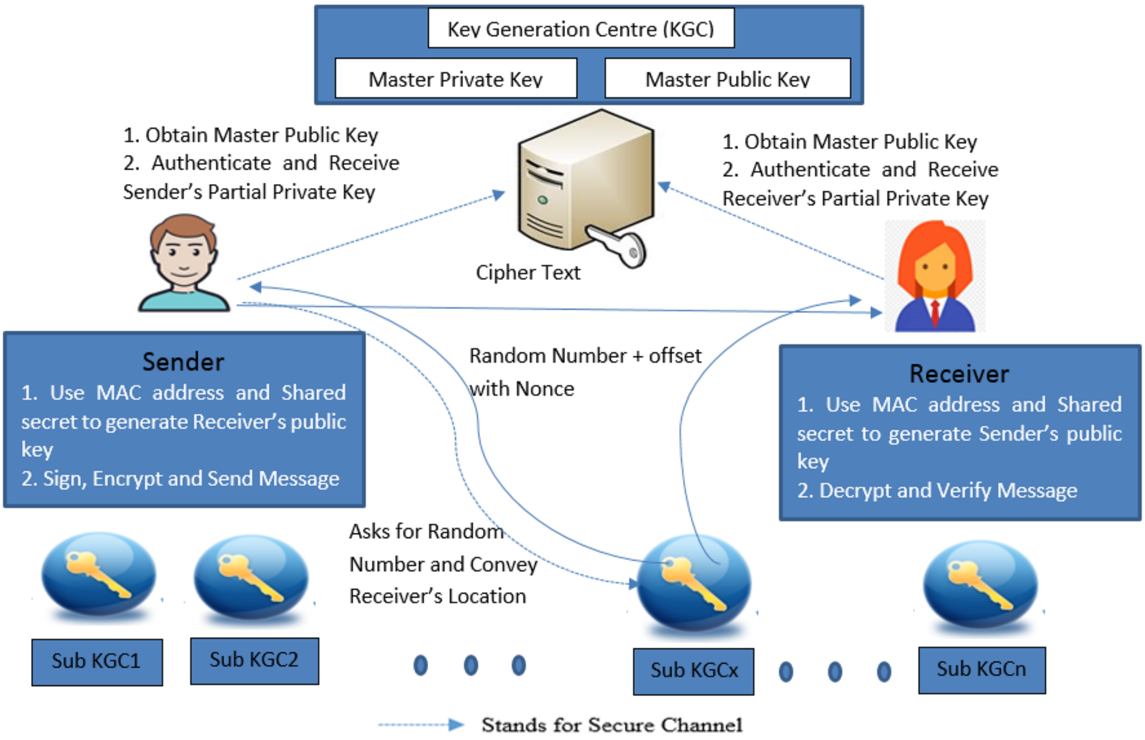

Key Generation Centre (KGC)
Master Private Key
Master Public Key
1. Obtain Master Public Key
2. Authenticate and Receive Sender's Partial Private Key
1. Obtain Master Public Key
2. Authenticate and Receive Receiver's Partial Private Key
Cipher Text
Random Number + offset with Nonce
Sender
1. Use MAC address and Shared secret to generate Receiver's public key
2. Sign, Encrypt and Send Message
Receiver
1. Use MAC address and Shared secret to generate Sender's public key
2. Decrypt and Verify Message
Asks for Random Number and Convey Receiver's Location
Sub KGC1
Sub KGC2
Sub KGCx
Sub KGCn
Stands for Secure Channel

Figure 3. Improved IBE System without Key-Escrow Problem (IIBES).

Figure 3 shows working of IIBES i.e. how the shared secret will be shared among two users by the use of sKGC. The IIBES tries to overcome the inherent key escrow problem of BF-IBE scheme and tries to provide better key revocation mechanism than RIBE scheme [16] by use of modified KUNode algorithm to select sKGC, by introducing the use of a nonce in shared secret and by use of Adi Shamir's secret sharing algorithm to deliver shared secret. IIBES also provides digital signature to prove authenticity of the user. In IIBES, we are going to use modified Shamir's signature scheme. Key Generation part is divided into two parts.

The first part is generating master public-private key pair as below:

Master public key – an RSA public key (n, e)

Where, n = pq (p and q are two large primes) and e is an arbitrary integer which is less than n and relatively prime to (p-1)*(q-1)

Master Private Key – an RSA private key (n, d)

$d = e^{-1} \bmod (p\text{-}1)*(q\text{-}1)$

The second part is the generation of client public-private key pair as below:

The public key for the user - his identity:

ID = MAC address || Shared Secret

Where a Shared secret is a random number and offset with a nonce.

The private key for the user:

$SId = ID^{d} \bmod n$

We assumed that all Local DNS and Client have computed the public-private key pair and the public key of all others are known to Local DNS and Client. When the client browses any website, NS of that will be provided by the client.

Generated query ID for our request will get encrypted and will be sent to the Local DNS as shown in Figure 4(a) by use of Equation 1:

$$t_C = r_C^e \ mod\ n$$
$$s_C=[SId_C*r_C^{H(tc,Id1)}]\ mod\ n \qquad (1)$$
$$EId_1 = (s_C,\ t_C)$$

Here (e, n) is a master public key, H() is a hash function (SHA-2 256), rc is a random number and Id1 is query Id i.e. message to be encrypted.

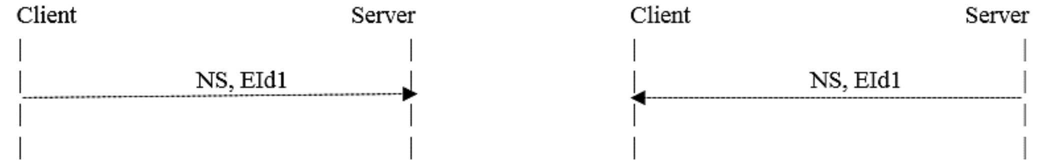
On receiving a request from client, Local DNS search respective IP for requested NS in database/file. If IP is found for requested NS, Local DNS encrypt the searched IP to send it to client as shown in Figure 4(b) by use of Equation 2:

$$t_L = r_L^e \ mod\ n$$
$$s_L=[SId_L*r_L^{H(tL,Id1)}]\ mod\ n \qquad (2)$$
$$EIP_1 = (s_L,t_L)$$

Here (e, n) is the public key of the Local DNS, H() is a hash function(SHA-2 256), r_L random number and Id1 is the response of Local DNS i.e. IP address for requested NS.

Local DNS also verifies the signature by calculating another signature at Local DNS side and then retrieve Id1 to send it along with EIP as seen in Equation 3:

$$s^e_c = [\,ID_c * t_c^{H(tc,Id1)}]\ mod\ n \qquad (3)$$



(a) Message sent from Client to Server

(b) Message sent from Server to Client

Figure 4. Client – Server Message Exchange.

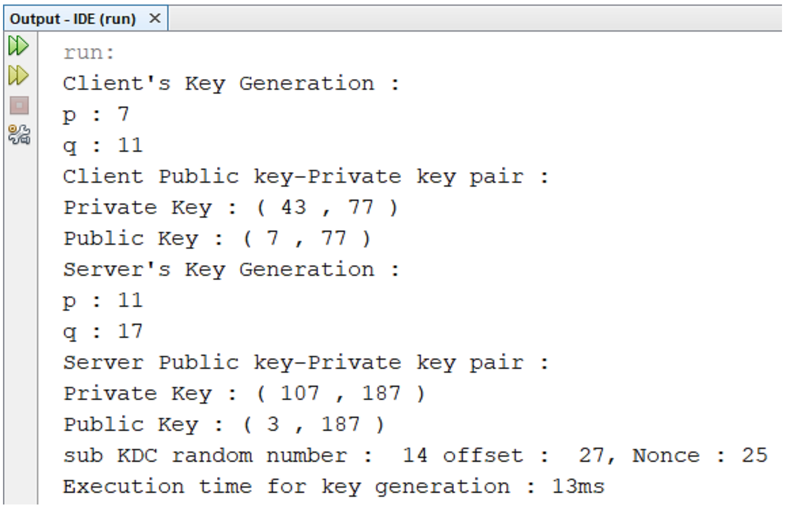
On receiving a message from Local DNS, the Client will match ID1 retrieve from Local DNS with Id1 which it has. If matched, the client will verify the signature by computing a new signature as seen in Equation 4:

$$s^e = [\,ID_L * t_L^{H(tL,Id1)}]\ mod\ n \qquad (4)$$

Using received IP address, client browses the website.

4- Results and Discussions

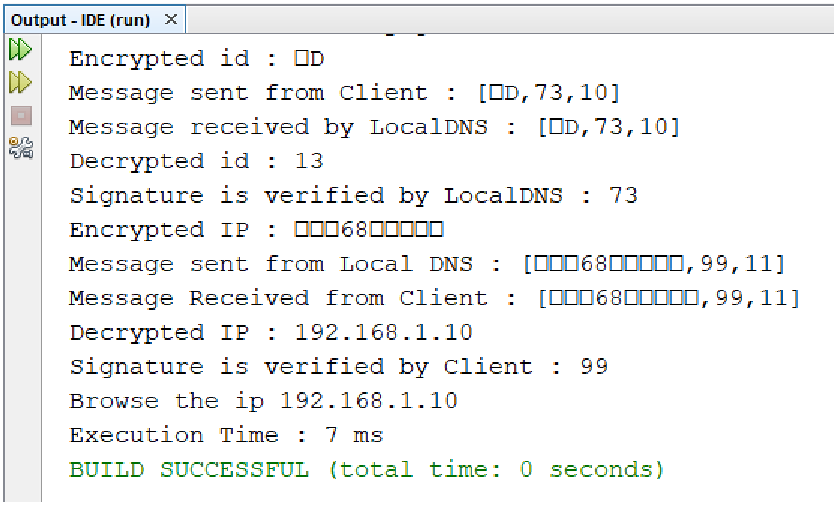
This section represents the generated outcome for the IIBES. The IIBES has been implemented using JAVA programming language. Figure 5 shows the generated public – private key pair for client and server using IIBES. The figure also displays value of random number, offset and nonce generated by sKGC. Execution time for generating public – private key pair has also been displayed in the figure.



```
run:
Client's Key Generation :
p : 7
q : 11
Client Public key-Private key pair :
Private Key : ( 43 , 77 )
Public Key : ( 7 , 77 )
Server's Key Generation :
p : 11
q : 17
Server Public key-Private key pair :
Private Key : ( 107 , 187 )
Public Key : ( 3 , 187 )
sub KDC random number :  14 offset :  27, Nonce : 25
Execution time for key generation : 13ms
```

Figure 5. Client and Server's Public – Private Key Generation.

Client encrypts the query id by using public key of server. Also, he/she performs digital signature by using his/her private key to provide authentication. Server will decrypt the IP by using private key and also verifies the signature by using client's public key. Upon finding the match for requested IP address, server encrypts the IP address and sends that encrypted IP address to the client with its signature. After verifying the signature, client browses website by using received IP address as shown in Figure 6.



```
Encrypted id : □D
Message sent from Client : [□D,73,10]
Message received by LocalDNS : [□D,73,10]
Decrypted id : 13
Signature is verified by LocalDNS : 73
Encrypted IP : □□□68□□□□□
Message sent from Local DNS : [□□□68□□□□□,99,11]
Message Received from Client : [□□□68□□□□□,99,11]
Decrypted IP : 192.168.1.10
Signature is verified by Client : 99
Browse the ip 192.168.1.10
Execution Time : 7 ms
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 6. Encryption and Decryption Process at Client-Server Side.

5- Conclusion

With the increasing use of Internet, there is a need of security system or protocol for the data transferred by DNS which can provide confidentiality, integrity and authenticity. IBE continues to increase in popularity over public key encryption as it reduces the overhead for certificate management by use of identity of users as public key. But, IBE system has its drawback like key escrow and key revocation problem. In this paper, a new scheme IIBES has been proposed in order to provide security to DNS using IBE. The IIBES provides confidentiality and authentication by use of IBE and IBS respectively for the data transferred by DNS. The IIBES significantly improves the key escrow and key revocation problem of IBE system by use of sKGC and shared secret with nonce. The IIBES can be compared with various candidate algorithms for different IBE elements to discover optimal performance in future. The IIBES also intended to be extended in terms of security attacks.

6- Declarations

6-1- Data Availability Statement

The data presented in this study are available in article.

6-2- Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

6-3- Conflicts of Interest

The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

3- References

[1] Chetioui, Kaouthar, Ghizlane Orhanou, and Said El Hajji. "Encryption of Query in DNS Message." International Journal of Security and Its Applications 9, no. 12 (December 31, 2015): 313–322. doi:10.14257/ijsia.2015.9.12.29.

[2] Ganji, Hamid Reza, and Kiarash Aghakhani. "Provides a New Way to Enhance Security in the Linux Operating System." Emerging Science Journal 2, no. 5 (November 4, 2018): 295. doi:10.28991/esj-2018-01153.

[3] Ariyapperuma, Suranjith, and Chris J. Mitchell. "Security Vulnerabilities in DNS and DNSSEC." The Second International Conference on Availability, Reliability and Security (ARES'07) (2007). doi:10.1109/ares.2007.139.

[4] Hussain, Mohammed Abdulridha, Hai Jin, Zaid Alaa Hussien, Zaid Ameen Abduljabbar, Salah H. Abbdal, and Ayad Ibrahim. "DNS Protection Against Spoofing and Poisoning Attacks." 2016 3rd International Conference on Information Science and Control Engineering (ICISCE) (July 2016). doi:10.1109/icisce.2016.279.

[5] Tsai, Tung-Tso, Yuh-Min Tseng, and Tsu-Yang Wu. “RHIBE: Constructing Revocable Hierarchical ID-Based Encryption from HIBE.” Informatica 25, no. 2 (January 1, 2014): 299–326. doi:10.15388/informatica.2014.16.

[6] Chen, Liqun, and John Malone-Lee. “Improved Identity-Based Signcryption.” Lecture Notes in Computer Science (2005): 362–379. doi:10.1007/978-3-540-30580-4_25.

[7] Elashry, Ibrahim, Yi Mu, and Willy Susilo. “Identity-Based Mediated RSA Revisited.” 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (July 2013). doi:10.1109/trustcom.2013.88.

[8] Lai, Jianchang, Yi Mu, Fuchun Guo, and Willy Susilo. “Improved Identity-Based Online/Offline Encryption.” Lecture Notes in Computer Science (2015): 160–173. doi:10.1007/978-3-319-19962-7_10.

[9] Boneh, Dan, Xuhua Ding, and Gene Tsudik. "Identity-based mediated RSA." In 3rd Workshop on Information Security Application, Jeju Island, Korea, vol. 12. 2002.

[10] Gagné, Martin. “Identity-Based Encryption.” Encyclopedia of Cryptography and Security (n.d.): 280–282. doi:10.1007/0-387-23483-7_193.

[11] Hu, Ziyuan, Shengli Liu, Kefei Chen, and Joseph Liu. “Revocable Identity-Based Encryption and Server-Aided Revocable IBE from the Computational Diffie-Hellman Assumption.” Cryptography 2, no. 4 (October 23, 2018): 33. doi:10.3390/cryptography2040033.

[12] Waters, Brent. “Efficient Identity-Based Encryption without Random Oracles.” Advances in Cryptology – EUROCRYPT 2005 (2005): 114–127. doi:10.1007/11426639_7.

[13] Kiltz, Eike, and Gregory Neven. 2008. Identity-based signatures. Vol. 2, chap. III in Cryptology and Information Security Series, by M. Joye and G. Neven, 31-44. IOS Press. doi:10.3233/978-1-58603-947-9-31.

[14] Soujanya, M. A., and Lalitha L. A. “An Efficiently Subcontracted the Identity-Based Encryption for Revocation in Cloud Computing.” International Journal of Science and Research (IJSR) 5, no. 5 (May 5, 2015): 514–520. doi:10.21275/v5i5.nov163317.

[15] Liu, Shengli, Yu Long, and Kefei Chen. “Key Updating Technique in Identity-Based Encryption.” Information Sciences 181, no. 11 (June 2011): 2436–2440. doi:10.1016/j.ins.2011.01.022.

[16] Gupta, Swati, and Vipul Gupta. “Revocable Key Identity Based Cryptography without Key Escrow Problem.” 2016 International Conference on Computing, Communication and Automation (ICCCA) (April 2016). doi:10.1109/ccaa.2016.7813817.

[17] Boldyreva, Alexandra, Vipul Goyal, and Virendra Kumar. “Identity-Based Encryption with Efficient Revocation.” Proceedings of the 15th ACM Conference on Computer and Communications Security - CCS ’08 (2008). doi:10.1145/1455770.1455823.

[18] Yuen, Tsz Hon, Willy Susilo, and Yi Mu. “How to Construct Identity-Based Signatures without the Key Escrow Problem.” International Journal of Information Security 9, no. 4 (July 22, 2010): 297–311. doi:10.1007/s10207-010-0110-5.

[19] Wei, Quanyun, Fang Qi, and Zhe Tang. “Remove Key Escrow from the BF and Gentry Identity-Based Encryption with Non-Interactive Key Generation.” Telecommunication Systems 69, no. 2 (May 14, 2018): 253–262. doi:10.1007/s11235-018-0461-1.

[20] Yuen, Tsz Hon, Cong Zhang, Sherman S. M. Chow, and Joseph K. Liu. “Towards Anonymous Ciphertext Indistinguishability with Identity Leakage.” Lecture Notes in Computer Science (2013): 139–153. doi:10.1007/978-3-642-41227-1_8.

[21] Lee, B., C. Boyd, E. Dawson, K. Kim, J. Yang and Seungjae Yoo. “Secure Key Issuing in ID-based Cryptography.” ACSW (2004).

[22] Seo, Jae Hong, and Keita Emura. “Revocable Identity-Based Encryption Revisited: Security Model and Construction.” Lecture Notes in Computer Science (2013): 216–234. doi:10.1007/978-3-642-36362-7_14.

[23] Alapati, Kalyan Koushik. “Group-oriented secret sharing using Shamir's algorithm.” (2018). doi:10.7282/T3K077G6.

[24] Shamir, Adi. “How to Share a Secret.” Communications of the ACM 22, no. 11 (November 1979): 612–613. doi:10.1145/359168.359176.

[25] Shamir, Adi. “Identity-Based Cryptosystems and Signature Schemes.” Lecture Notes in Computer Science (November 2000): 47–53. doi:10.1007/3-540-39568-7_5.