CLIKEV2: A Novel Key Agreement Protocol for Securing and Managing Biomedical Data in Iot

Lavanya M.* and Natarajan V.
Research Scholar, MIT Campus, Anna University, Chennai, INDIA
*cmbt8349@gmail.com

Abstract

The Internet of Things (IoT) communication is now widely used for e-health communication as it allows heterogeneous hardware and services to merge seamlessly in providing an efficient healthcare system. As sensitive and private biomedical information is communicated, the IoT system must provide security for the data. The most popular security mechanisms are encryption and key management. The key management schemes of IoT should be based on standard Internet protocols to communicate in a heterogeneous environment. The standard IPSec protocol relies on IKE for key agreement. IKEv2 is a minimal version of IKE suitable for IoT. IKEv2 provides certificate based authentication and Diffie Hellman Key agreement scheme which is too heavy for IoT. Hence in this paper a lightweight, certificate-less protocol based on ECC is proposed. This protocol uses Elliptic curve Diffie Hellman key exchange instead of the RSA Diffie Hellman used in IKEv2. The major advantages of the proposed scheme are that the security level is same as RSA with reduced key sizes, ECC based key agreement allows to generate a common shared key, which can be used for Symmetric encryption techniques which requires less processing time, storage and computation cost.

Introduction

The term "Internet of Things" (IoT) coined by Kevin Ashton describes about how the objects present in physical world connected through sensors to the Internet. Press department and popular media put headline news about IoT as it is a vital topic in technology industry, policy, and engineering circles. This technology benefits from the developments in computing power, miniaturization of electronic devices, and wireless networks for providing new capabilities which is not possible previously. This is specifically seen in biomedical applications, where compact body sensors are used for measuring the vital signs and the biomedical data is transmitted; thus enabling medical professionals to monitor remotely. Profusion of conferences, reports, and news articles are mainly made a discussion and debate on future impact of the "IoT revolution" as of new market opportunities and business models that apprehensions around security, privacy, and technical interoperability [1].

The Internet today is becoming an IPv6 network connecting to smart objects and networks like wireless sensor networks along with the traditional computers. This IoT will provide many services to meet our day to day life. Due to the development of low cost, low power wearable medical sensors and equally advanced communication technology has made e-health care solutions viable through internet. The advent of IoT has made possible for individual health data collected by body sensors and the biomedical data transmitted to the physicians for monitoring, analyzing and diagnosing [2]

IoT implementation includes layers such as data acquisition present below where application layer presented at the top. This layered architecture met necessities of various industries, enterprises, societies, institutes, governments etc. IoT's generic layered architecture is represented in figure 1 which has two different divisions with Internet layer present in between. For communication, this assists the purpose of a common media [3] For data capturing, two lower layers contribute and the two top layers are responsible for data utilization.

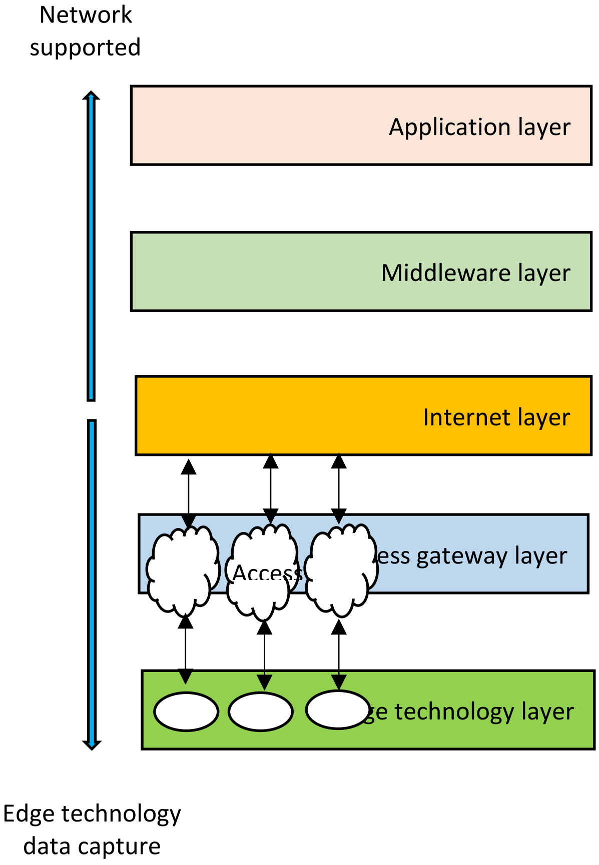


Figure 1: Layered architecture of Internet of Things

Functionalities of different layers are explained in short:

- **Edge layer**: This is a hardware layer made up of or networks, embedded systems, RFID tags and readers or other soft sensors. This layer basically contains the data sensors which offer identification and information storage. In a biomedical application, the sensors are used detection vital signs of patients.
- **Access gateway layer**: Data handling's first stage is done on this layer. It gives attention on message routing, publishing and subscribing. If needed, cross platform communication is provided.
- **Middleware layer**: This operates on bidirectional mode which acts on an interface between the hardware layer present at the bottom and an application layer present at the top of the layer. For serious functions, it is responsible like device management and information management.
- **Application layer**: This is presented at the top of the stack and in IoT, it delivers various applications to different users. Various industry verticals provide applications such as manufacturing, logistics, retail, environment, public safety, healthcare, food and drug etc. various applications are developed with increase of maturity of RFID technology under the umbrella of IoT.

IoT devices is implemented as large-scale which possibilities to transform based on its feature and way of living. New IoT products are moved towards "smart home'' vision for consumer like Internet-enabled appliances, home automation components, and energy management devices which offers high security and energy efficiency. Wearable fitness, health monitoring devices and medical devices that are enabled by network delivers the way healthcare services. People with disabilities benefited by this technology and independence has been improved with improving life quality at a reasonable cost. IoT systems are moved to smart city idea such as networked vehicles, intelligent traffic systems, and embedded sensor in roads and bridges. This minimizes congestion and energy consumption. Also, it offers transform agriculture, industry, and energy production. It increases the information availability with value chain of production using networked sensors [4].

IoT technology is applied to application of "environmental monitoring" by sensing ability, distributed and self-managing fashion, natural phenomena and processes like temperature, wind, rainfall, river height and merge this heterogeneous data into global applications. IoT technologies monitors availability of "real-time product" in trade application and accurate stock inventory is maintained. Same type of threats is faced by homeland security that albeit on a different scale. This technology enhances the performance of current solutions and it provides a cheap and less offensive alternative for the developments of cameras and to preserve users' privacy.

Hence there is a need for providing secure communication in IoT. The various security services like confidentiality, authentication, integrity, replay protection become mandatory for biomedical applications to secure the data. The standard protocols provide security at different layers. The table.1 gives comparison of different layers and security protocols used in IoT and the general Internet applications.

Table 1
Layers and Security Solutions

Layer	IOT +Security	General Internet Application + Security
Application	CoAP+CoAPs	HTTP + HTTPs
Transport	UDP + DTLS	TCP + TLS
Network	IPv6, RPL,6LoWPAN+ IPSec, RPL security	IPv4, IPv6 + IPSec
Data Link	IEEE802.15.4 + IEEE802.15.4 Security	IEEE802.3, IEEE802.11

Using standardized Internet security solutions, it is possible to provide secure communication at different layers as shown in table.1. In the link layer 6LowPAN networks are secured using IEEE 802.15.4 security. This solution provides security in per hop basis, where every node in the network has to be trusted. It provides security using single pre-shared key for the entire network, which is the disadvantage. Though IoT is a developing technology rapidly, some suspicions about its security and privacy occurs which affect its bearable development. Farooq et al[5] examined issues and challenges of security and security architecture was provided as a confidentiality of the user's privacy and security that results in its wider adoption by masses.

Huang et al., [6] addressed on problem of malicious attacks through original security framework development with strong and transparent security protection. Based on three characteristic investigation was made into the security requirements such as IoT scenarios, new authentication design and an access control subsystem using fine-grained roles and risk indicators. Insight was provided by security framework on main difficulties of IoT security and on providing some feasible solutions.

Nia & Jha [7] presented complete vulnerability list and countermeasures in contradiction of them on the edge-side layer of IoT that consists of three different levels such as edge nodes, communication and edge computing. Initially brief discussion on IoT reference models was made and security was defined in the context of IoT. Then IoT applications were discussed and attacker's potential motivations target this new paradigm. Discussion was made on different attacks and threats. Possible countermeasures have been described against these attacks and finally two emerging security challenges was introduced.

Security in network layer is provided by IPSec. IPSec in transport mode provides end to end security including authentication and replay protection. IPSec can be used with CoAP. IPSec [8] is mandatory in IPv6 protocol. Even though IPSec is used in IoT, it is not designed specifically for web protocols like HTTP or CoAP. TLS and SSL provide solution in traditional web protocols, TLS can be used over TCP and for UDP DTLS is available CoAPs mandates the use of DTLS, hence it is necessary to enable DTLS for IoT. IPSec is application independent. It works with protocols like Authentication Header [2], Encapsulated security payload (ESP) [9]and Internet key Exchange (IKE) [10]. IKE protocol has some limitations because of the cookies and nonces as discussed in [11] As a solution to this IKEv2 [22], SIGMA [12] and JFK [13] are coined.

IKEv2 is based on public key signatures. It hides the identities from the passive attackers and the number of message exchanged is reduced. IKEv2 uses Diffie Hellman[17] key exchange and uses RSA Signatures, which is less secure and heavy key length. The other certificate less schemes discussed in [14,15] are based on bilinear parring, which has a very heavy processing overhead.

To overcome these difficulties a novel certificate-less key agreement scheme based on ECC [16] is proposed in this paper. It replaces D-H key exchange with ECDH [18] key exchange also the usage of Certificates for authenticating the public keys is eliminated in this scheme. An IKE protocol for Internet applications with ECC has been discussed in [19] where prfs (pseudo random functions) are used for calculating the authenticating element, which is replaced by one way sponge based hash function [20] in the proposed protocol. A collaborative key management approach is discussed in [21], where intermediate proxies are used, who will generate the signature on behalf of the initiator and verify on behalf of the responder. The communication overhead is increased in this approach compared to the proposed protocol.

The paper is organized as follows, Section 2 explains the existing IKEv2 protocol and its key agreement scheme in detail, Section 3 explains the proposed key agreement protocol for IoT, Section 4 analyses the security of the proposed protocol and Section 5 concludes the work.

IKEv2 Protocol - A Review

IKE is a security protocol used for establishing security association(SA) for IPSec. The security mechanism like the cryptographic algorithms hash functions and the keying materials agreed upon by the IPSec peers is called the Security association. The protocol is a request response type with an initiator and a responder.

IKE has two versions IKEv1 and IKEv2 [22]. There are two phases in IKEv2:

Phase I: Mutual authentication, Session key establishment, it also establishes an IKE SA for the next phase.

Phase II: Using the Phase I SA, Multiple Phase II SAs are generated between the same peers for further communication, these are called as CHILD SA or IPSec SA.

The Phase I and II message exchanges are shown in the figure.2. The notations and the corresponding definitions for the terms used are discussed in the table 2.

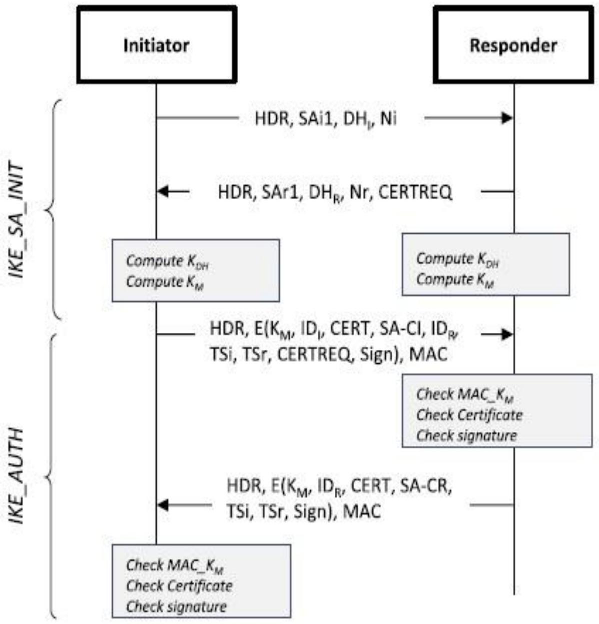


Figure 2: L IKEv2 BASIC protocol

There are two types of phase I exchanges: Aggressive mode and the Main mode. In aggressive mode, the authentication and key establishment are done with three message exchanges, while in the main mode it takes six messages with additional functionalities. The first request/response of the session i.e. phase I is called as IKE_SA_INIT, this session negotiates security parameters for next phase IKE_AUTH. This phase generates AH or ESP CHILD_SA. IKE_SA_INIT sends nonces and DH values and computes the session key. IKE_AUTH transmits identities, proves that the corresponding entities knows the secrets, and sets up CHILD_SA.

CLIKEv2: Proposed Certificate-less IKEv2 protocol

The proposed protocol eliminates the use of certificates and cookies from the traditional protocol. The protocol is ECC based hence the domain parameters of the elliptic curves are agreed upon by both communicating entities. The initiator and the responder decide upon their private keys and generate corresponding public keys. Here the standard elliptic curve operating on prime field is used for implementation. The notations and symbols used in the proposed algorithm is same the one used in the traditional algorithm some of the parameters that was not defined in

IKEv2 and are specific for CLIKEv2 are listed in the table.3. The initiator must know the ID and the IP of the responder and vice versa.

Table 2
Notations and symbols

HDR	**General IKE Header**
SA	Security Association
i, r	Initiator, Responder
KE	Key Exchange
N	Nonce
CERTREQ	Certificate request
CERT	Certificate
ID	Identification
AUTH	Authentication
TS	Traffic Selector

Table 3
Notations and symbols

Flg_r	**Digest to authenticate r**
Pu_r PU_i	Public key of i,r
i,r	Initiator, Responder
k	Session key
k_x, ky	Session key elements
IPi, IPr	IP of i, r
H1, H2, H3	Message digests

Phase-I of CLIKEv2: The details of the phase-I of the proposed protocol shown in figure.3. is as follows:

Step 1: Initiator **-->** Responder}: HDR, SAi

The initiator sends the offered cryptographic solutions to the responder.

Step 2: Responder**-->** Initiator}: HDR, SAr, PUr

The responder calculates a digest, Flgr = h(IPr,IDr,PUr), and selects the required cryptographic algorithms from the offered solutions and sends it along with its public key and nonce. The nonce prevents the replay attack. If the responder is not satisfied with the offered cryptographic solutions it can send an error message in this stage.

Step 3: Initiator **-->**Responder: HDR,Pui,Flg_i,E{kx}(Nr || Ni)

The initiator computes its own Flgr and compares it with the received value. If there is a mismatch the communication is terminated, otherwise the initiator calculates the session key from the received public key of responder and generates its own Flgi and transmits this value of digest, its public key, and the nonce of sender and receiver along with its IP encrypted with the key component of the session key. Hence responder is verified and replay is checked.

Step 4: Responder **-->**Initiator}: E{kx}(Nr||Ni||IPr)

The responder verifies the digest of initiator Flgi , if it is true calculates the session key using public key of responder. It sends the encrypted values of the nonces to initiator for mutual authentication.

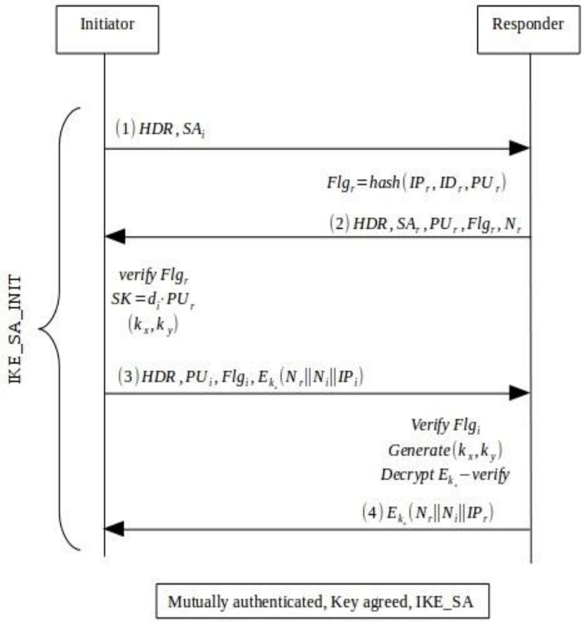

Initiator
Responder
IKE_SA_INIT
(1) HDR, SA_i
$Flg_r=hash(IP_r, ID_r, PU_r)$
(2) $HDR, SA_r, PU_r, Flg_r, N_r$
verify Flg_r
$SK=d_i \cdot PU_r$
(k_x, k_y)
(3) $HDR, PU_i, Flg_i, E_{k_x}(N_r||N_i||IP_i)$
Verify Flg_i
Generate(k_x, k_y)
Decrypt E_{k_x} − verify
(4) $E_{k_x}(N_r||N_i||IP_r)$
Mutually authenticated, Key agreed, IKE_SA

Figure 3: CLIKEv2 Phase I

After successful mutual authentication, the initiator and responder agree to calculate derived keys like authentication key, encryption key, integrity key from the session key elements

Phase-II of CLIKEv2

Phase-II operates in quick mode. Either the initiator or the responder can initiate this phase. New public keys are generated if needed. All messages in quick mode are encrypted with phase-I SA's encryption key and integrity protected with the hash values. quick mode need not include the identities of the initiator and the responder since they are authenticated in phase-I. The details of phase-II shown in figure 4.\ref{p2} is as follows.

Step 1: Initiator-->Responder}: HDR,E{Kx}(H1,SA,Ni)
The initiator generates the hash value H1 as H1=h(ky,SA,Ni) for authentication , the SA is the phase-I SA and Ni is the nonce generated in this phase. this hash value is encrypted with the session key element of phase-I and transmitted to the responder.

Step 2: Responder **-->**Initiator}: HDR,E{kx}(H2,SA,Nr

The responder calculates H1 with the received values of SA, Ni and the known session key value and compares this H1 with the received H1. If they match the responder calculates H2 as H2 = h(ky,SA,Nr) to authenticate the message where n_r is the new nonce generated for phase-II. Now it encrypts this nonce and transmits to the initiator.

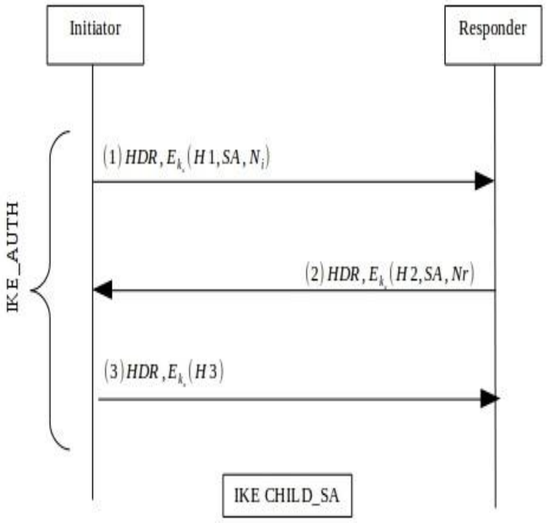


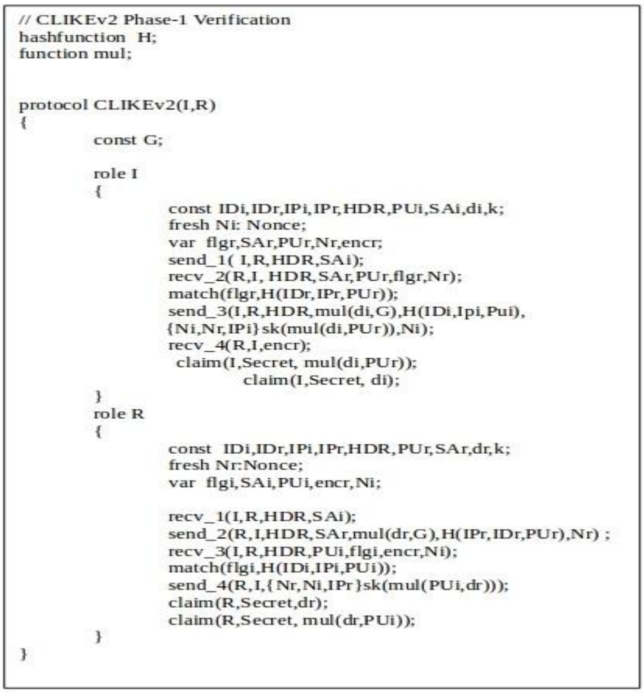
Figure 4: CLIKEv2 Phase -II

Step 3: Initiator –>Responder}: HDR, E_{k_x}(H3)

The responder calculates its own H2 using the received value. if H2 matches H3 is calculated, and transmitted to the responder.

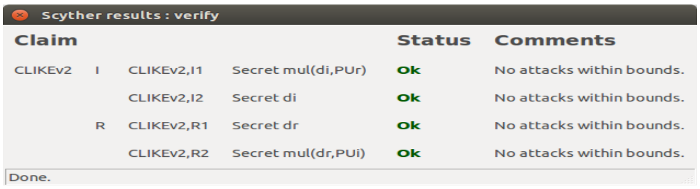
Protocol Verification

The proposed protocol CLIKEv2 is analysed using the automatic protocol verification tool Scyther. The security properties like secrecy, authentication characterized in terms of aliveness, synchronization and message agreement are verified using trace patterns. The trace pattern helps in capturing the class of all attack traces for a given protocol and security property. If there are any attack patterns in any specific traces, then it is concluded that the security property is false. The programming language for Scyther is security protocol definition language(spdl). The spdl code for phase-I of CLIKEv2 is shown in figure 6 & 9.\ref{scyther_p1}. The user defined and automatic claims are verified and shown in the figures 7 & 10 \ref{v1} and \ref{v12} respectively. Similarly, the spdl code for phase-II is shown in figure.\ref{scyther_p2} and the corresponding claim verifications are shown in the figures 5 & 8 \ref{v2} and \ref{v21}. When the verification claims are analysed it can be concluded that the proposed protocol has no attacks defined in both the phases. The protocol gives good results in both manual and automatic claims in both the phases.



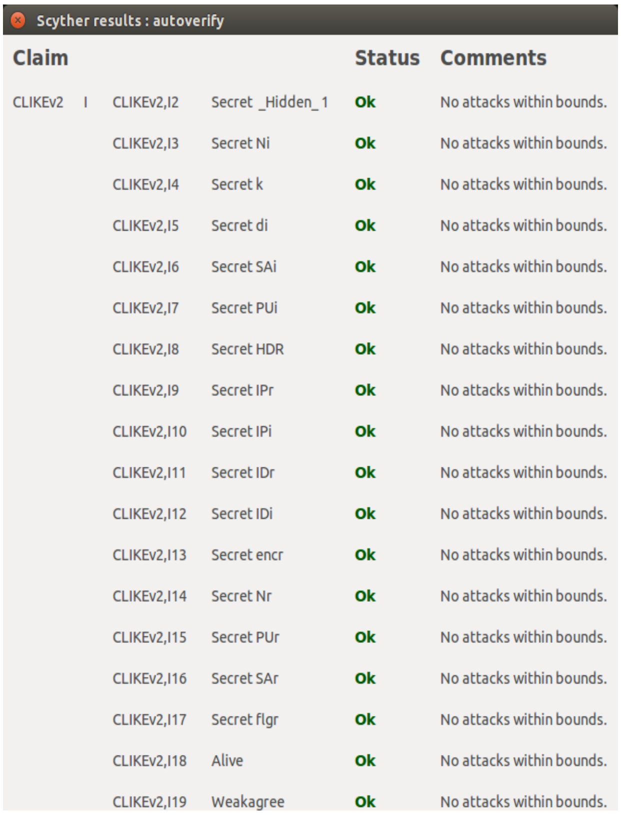
```
// CLIKEv2 Phase-1 Verification
hashfunction  H;
function mul;


protocol CLIKEv2(I,R)
{
        const G;

        role I
        {
                const IDi,IDr,IPi,IPr,HDR,PUi,SAi,di,k;
                fresh Ni: Nonce;
                var  flgr,SAr,PUr,Nr,encr;
                send_1( I,R,HDR,SAi);
                recv_2(R,I, HDR,SAr,PUr,flgr,Nr);
                match(flgr,H(IDr,IPr,PUr));
                send_3(I,R,HDR,mul(di,G),H(IDi,Ipi,Pui),
                {Ni,Nr,IPi}sk(mul(di,PUr)),Ni);
                recv_4(R,I,encr);
                 claim(I,Secret, mul(di,PUr));
                         claim(I,Secret, di);
        }
        role R
        {
                const  IDi,IDr,IPi,IPr,HDR,PUr,SAr,dr,k;
                fresh Nr:Nonce;
                var  flgi,SAi,PUi,encr,Ni;

                recv_1(I,R,HDR,SAi);
                send_2(R,I,HDR,SAr,mul(dr,G),H(IPr,IDr,PUr),Nr) ;
                recv_3(I,R,HDR,PUi,flgi,encr,Ni);
                match(flgi,H(IDi,IPi,PUi));
                send_4(R,I,{Nr,Ni,IPr}sk(mul(PUi,dr)));
                claim(R,Secret,dr);
                claim(R,Secret, mul(dr,PUi));
        }
}
```

Figure 5: CLIKEv2 Phase I SPDL code



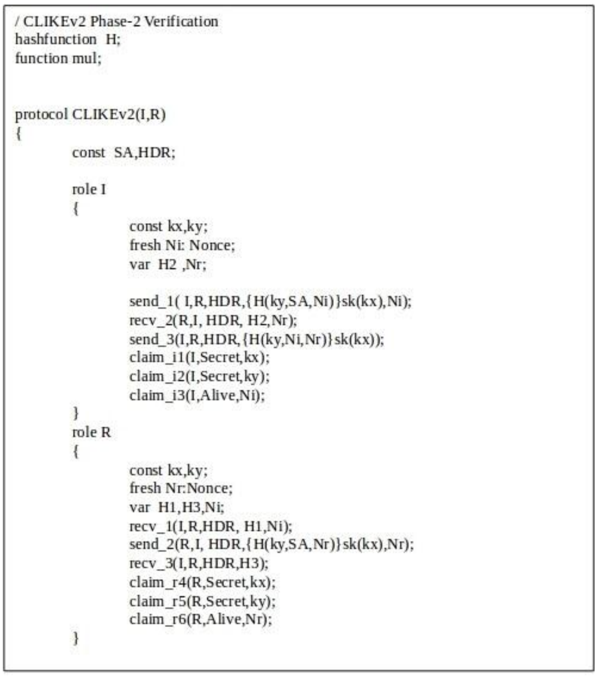
```
Claim                                                   Status   Comments
CLIKEv2    I    CLIKEv2,I1    Secret mul(di,PUr)        Ok       No attacks within bounds.
                CLIKEv2,I2    Secret di                 Ok       No attacks within bounds.
           R    CLIKEv2,R1    Secret dr                 Ok       No attacks within bounds.
                CLIKEv2,R2    Secret mul(dr,PUi)        Ok       No attacks within bounds.
Done.
```

Figure 6: CLIKEv2 Phase I Claim verification



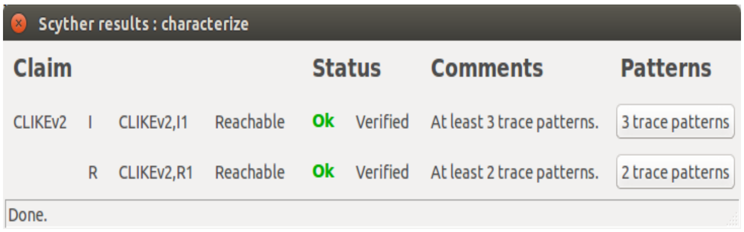
```
Claim                                               Status   Comments
CLIKEv2   I    CLIKEv2,I2     Secret _Hidden_1      Ok       No attacks within bounds.
               CLIKEv2,I3     Secret Ni             Ok       No attacks within bounds.
               CLIKEv2,I4     Secret k              Ok       No attacks within bounds.
               CLIKEv2,I5     Secret di             Ok       No attacks within bounds.
               CLIKEv2,I6     Secret SAi            Ok       No attacks within bounds.
               CLIKEv2,I7     Secret PUi            Ok       No attacks within bounds.
               CLIKEv2,I8     Secret HDR            Ok       No attacks within bounds.
               CLIKEv2,I9     Secret IPr            Ok       No attacks within bounds.
               CLIKEv2,I10    Secret IPi            Ok       No attacks within bounds.
               CLIKEv2,I11    Secret IDr            Ok       No attacks within bounds.
               CLIKEv2,I12    Secret IDi            Ok       No attacks within bounds.
               CLIKEv2,I13    Secret encr           Ok       No attacks within bounds.
               CLIKEv2,I14    Secret Nr             Ok       No attacks within bounds.
               CLIKEv2,I15    Secret PUr            Ok       No attacks within bounds.
               CLIKEv2,I16    Secret SAr            Ok       No attacks within bounds.
               CLIKEv2,I17    Secret flgr           Ok       No attacks within bounds.
               CLIKEv2,I18    Alive                 Ok       No attacks within bounds.
               CLIKEv2,I19    Weakagree             Ok       No attacks within bounds.
```

Figure 7: CLIKEv2 Phase I Automatic Claim verification



```
/ CLIKEv2 Phase-2 Verification
hashfunction  H;
function mul;


protocol CLIKEv2(I,R)
{
        const  SA,HDR;

        role I
        {
                const kx,ky;
                fresh Ni: Nonce;
                var  H2 ,Nr;

                send_1( I,R,HDR,{H(ky,SA,Ni)}sk(kx),Ni);
                recv_2(R,I, HDR, H2,Nr);
                send_3(I,R,HDR,{H(ky,Ni,Nr)}sk(kx));
                claim_i1(I,Secret,kx);
                claim_i2(I,Secret,ky);
                claim_i3(I,Alive,Ni);
        }
        role R
        {
                const kx,ky;
                fresh Nr:Nonce;
                var  H1,H3,Ni;
                recv_1(I,R,HDR, H1,Ni);
                send_2(R,I, HDR,{H(ky,SA,Nr)}sk(kx),Nr);
                recv_3(I,R,HDR,H3);
                claim_r4(R,Secret,kx);
                claim_r5(R,Secret,ky);
                claim_r6(R,Alive,Nr);
        }
```

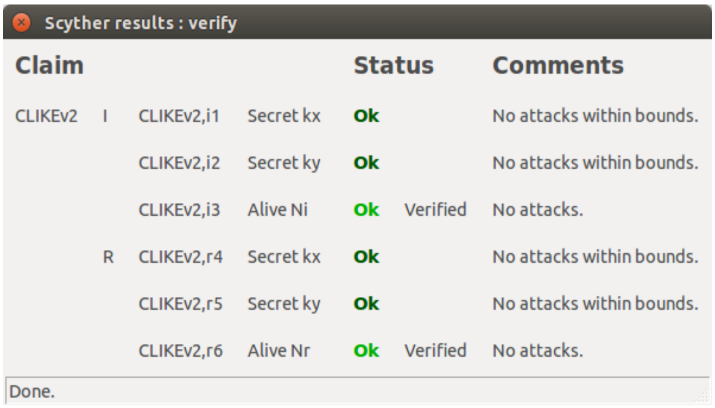
Figure 8: CLIKEv2 Phase II SPDL code



Claim				Status		Comments	Patterns
CLIKEv2	I	CLIKEv2,I1	Reachable	Ok	Verified	At least 3 trace patterns.	3 trace patterns
	R	CLIKEv2,R1	Reachable	Ok	Verified	At least 2 trace patterns.	2 trace patterns

Done.

Figure 9: CLIKEv2 Phase II Claim verification



Claim				Status		Comments
CLIKEv2	I	CLIKEv2,i1	Secret kx	Ok		No attacks within bounds.
		CLIKEv2,i2	Secret ky	Ok		No attacks within bounds.
		CLIKEv2,i3	Alive Ni	Ok	Verified	No attacks.
	R	CLIKEv2,r4	Secret kx	Ok		No attacks within bounds.
		CLIKEv2,r5	Secret ky	Ok		No attacks within bounds.
		CLIKEv2,r6	Alive Nr	Ok	Verified	No attacks.

Done.

Figure 10: CLIKEv2 Phase II Automatic Claim verification

Security analysis: The following analysis helps in finding the strength of the proposed protocol.

Man-in-the-Middle attack: Assume if there is an intruder A between initiator I and the responder R. If A is able to generate a separate secret session keys with A and R , and is able to communicate with I and R then it is man-in-the-middle attack. A is now able to modify the messages I has sent to R and vice versa. In order to avoid this the $Flg_r and Flg_i$ fields of message 2 and 3 of phase-I include the IP address of R and I respectively. Hence the intruder will not be able to generate the same values for $Flg_r and Flg_i$ as generated by I and R hence avoiding man-in-the-middle attack.

Replay attack: In a replay attack an intruder can easily replay the values send by the actual communicating entities in later stage of the communication thereby confusing the entities communicating. this is avoided by the use of nonce in the responder side and also from the initiator side. Nonce is a number used one time it can be time stamp also.

Non-Repudiation: Neither Initiator nor the responder can deny that the message was not sent from their side because the $Flg_r a
nd Flg_i$ values sent by both has its own ID and the IP address together. Hence non-repudiation is overcome by the proposed protocol.

Overheads reduced: The infrastructural complexities like public key certificates, cookies, need for a certification authority to maintain the certificates are avoided in the proposed protocol. Hence it provides an efficient way of developing a public key infrastructure for key agreement.

Table 4
Comparison of Traditional and CLIKEv2 protocol

Parameters	Traditional IKEV2	Proposed CLIKEv2
Cryptographic technique used	RSA	ECC
Encryption	Public Key	Symmetric key
Method of Key agreement	D-H	ECDH
Cookies avoided	No	Yes
Key Size Less	No (1024 bits)	Yes (160 bits)
Computation Overhead	Yes	No
Avoids Man-in-middle, DOS attacks	Yes	Yes
Message payload reduced	No	Yes

Conclusion

Security is paramount in managing biomedical data in IoT. In biomedical applications, it is essential to maintain the confidentiality and privacy of the data. CLIKEv2 uses ECC for providing security, hence there is a reduction in the key sizes for the same level of security that RSA can provide. The overhead in the transmitted message is reduced by avoiding certain fields like CERTREQ, CERT, AUTH. This concludes that the proposed protocol is efficient compared to the RSA based protocols. A formal protocol verification is also done using the automatic protocol verification tool Scyther. From the analysis results it can be concluded that the protocol is well formed, and there are no trace of attacks in the protocol design. As a conclusion, a comparative result of the traditional and the proposed scheme is given in the table.\ref{comp}.

References

1. Rose, K., Eldridge, S., & Chapin, L, The internet of things: An overview. The Internet Society (ISOC), **1-50 (2015).**

2. Wei, J, How Wearables Intersect with the Cloud and the Internet of Things: Considerations for the developers of wearables. IEEE Consumer Electronics Magazine, **3(3), 53-56. (2014).**

3. Bandyopadhyay, D., & Sen, J, Internet of things: Applications and challenges in technology and standardization. Wireless Personal Communications, **58(1), 49-69 (2011)**.

4. Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I., Internet of things: Vision, applications and research challenges. Ad Hoc Networks, **10(7), 1497-1516. (2012)**

5. Farooq, M. U., Waseem, M., Khairi, A., & Mazhar, S, A critical analysis on the security concerns of internet of things (IoT). International Journal of Computer Applications, **111(7) (2015)**

6. Huang, X., Craig, P., Lin, H., & Yan, Z., SecIoT: a security framework for the Internet of Things. Security and Communication Networks **(2015)**

7. Nia, A. M., & Jha, N. K. (2016). A comprehensive study of security of internet-of-things. IEEE Transactions on Emerging Topics in Computing **(2016)**

8. Bui, N., & Zorzi, M ,Health care applications: a solution based on the internet of things. In Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (p. 131). ACM. **(2011)**

9. Kaufman, C., & Perlman, R,. Key exchange in IPSec: analysis of IKE. IEEE Internet Computing, 4(6), 50-56 **(2000).**

10. Harkins, D., & Carrel, D, The internet key exchange (IKE) (No. RFC 2409) **(1998)**.

11. Nagalakshmi, V., Babu, I. R., & Avadhani, P. S. Modified Protocols for Internet Key Exchange (IKE) Using Public Encryption and Signature Keys. In Information Technology: New Generations (ITNG), 2011 Eighth International Conference on (pp. 376-381). IEEE. **(2011).**

12. Krawczyk, H., SIGMA: The 'SIGn-and-MAc'approach to authenticated Diffie-Hellman and its use in the IKE protocols. In Annual International Cryptology Conference (pp. 400-425). Springer Berlin Heidelberg. **(2003)**.

13. Aiello, W., Bellovin, S. M., Blaze, M., Ioannidis, J., Reingold, O., Canetti, R., & Keromytis, A. D, Efficient, DoS-resistant, secure key exchange for internet protocols. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 48-58). ACM **(2002)**

14. Yaacob, A. H., Ahmad, N. M., Fauzi, R., & Shikh, M. S. A. M., IKE authentication using certificateless signature. In Information Networking (ICOIN), 2011 International Conference on **(pp. 447-452). IEEE (2011).**

15. Ahmad, N. M., Yaacob, A. H., Fauzi, R., & Khorram, A., Performance analysis of certificateless signature for IKE authentication. World Academy Science, Engineering and Technology, **74, 358-365 (2011)**.

16. Koblitz, N. Elliptic Curve Cryptosystem, Journal of mathematics computation, **48(177):203-2009. (1987)**.

17. ANSI X9.62, Elliptic Curve Key Agreement and Key Transport Protocols. American Bankers Association. **(1999)**.

18. Ray, S., Nandan, R., & Biswas, G. P, ECC based IKE protocol design for internet applications. Procedia Technology, **4, 522-529 (2012)**.

19. Lavanya, M., & Natarajan, V, Implementation of ECDSA Using Sponge Based Hash Function. In Computational Intelligence, Cyber Security and Computational Models (pp. 349-359). Springer Singapore **(2016)**.

20. Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M, Lightweight collaborative key establishment scheme for the Internet of Things. Computer Networks, **64, 273-295 (2014).**
2. Kaufman, C, Internet key exchange (IKEv2) protocol **(2005)**.

21. Cremers, C. J. F., Scyther: Semantics and verification of security protocols. Eindhoven, Netherlands: Eindhoven University of Technology **(2006)**.