Lino Zachariah, Dr. Poonam Sinha / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.043-047

Analysis and comparison of ECC & ECIES using IBE for securing patient's privacy

Lino Zachariah
M-Tech(IT), BU Bhopal

Dr. Poonam Sinha
HOD-Information Technology
BU, Bhopal (MP)

ABSTRACT

The work that we are presenting here denotes the comparison between Elliptic curve cryptography and elliptic curve integrated Encryption Scheme using Identity Based Encryption implemented on sensor networks, motivated by those networks' need for an efficient, secure mechanism for shared cryptographic keys' distribution and redistribution among nodes. Both the cryptographic technique can be implemented as an application for securing the patient's privacy in wireless body sensor networks. Hence, we are comparing the their implementation on body sensor network for the encryption and decryption of patient's private record.

INTRODUCTION

Body Sensor Networks (also known as bodynets or Body Area Networks) have the potential to revolutionize healthcare monitoring. These networks are comprised of wearable devices with attached sensors that can measure various physiological and environmental signals. Bodynet devices communicate wirelessly with networked gateways (mobile phones, computers and PDAs) which store, analyse and communicate vital information in real-time. A Bodynet can be designed to immediately alert emergency personnel to a critical situation like a heart attack or a debilitating fall. Bodynets can also help physicians catch warning signs of a disease earlier or remotely monitor the progress of a recovering surgery patient.

A Wireless Sensor Network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. The envisaged size of a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust although functioning 'motes' of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few cents, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors (including specific conditioning circuitry), a communication device (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user.

IDENTITY BASED ENCRYPTION

Identity-based encryption (IBE) is a public-key encryption technology that allows a public key to be calculated from an identity and a set of public mathematical parameters and for the corresponding private key to be calculated from an identity, a set of public mathematical parameters and a domain-wide secret value. An IBE public key can be calculated by anyone who has the necessary public parameters; a cryptographic secret is needed to calculate an IBE private key, and the calculation can only be performed by a trusted server which has this secret.
Identity-based systems allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the Private Key Generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as master key). Given the master public key, any party can compute a public key corresponding to the identity ID by combining the master public key with

the identity value. To obtain a corresponding private key, the party authorized to use the identity ID contacts the PKG, which uses the master private key to generate the private key for identity ID.

As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants. This is extremely useful in cases where pre-distribution of authenticated keys is inconvenient or infeasible due to technical restraints. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the PKG. A caveat of this approach is that the PKG must be highly trusted, as it is capable of generating any user's private key and may therefore decrypt (or sign) messages without authorization. Because any user's private key can be generated through the use of the third party's secret, this system has inherent key escrow. A number of variant systems have been proposed which remove the escrow including certificate-based encryption, secure key issuing cryptography and certificate less cryptography [25].
IBE scheme consists of four algorithms:

(1) **Setup** generates global system parameters and a master-key.
(2) **Extract** uses the master-key to generate the private key corresponding to an arbitrary public key string ID.
(3) **Encrypt** encrypts messages using the public key ID, and
(4) **Decrypt** decrypts messages using the corresponding private key.
(1) Setup: IBE systems rely upon a trusted central authority that manages the parameters with which keys are created. This authority is called the Private Key Generator or PKG. The PKG creates its parameters, including a master secret from which private keys are created.
(2) Extraction: When doctor wishes to decrypt the message C that was encrypted, he/she authenticates himself to the PKG and obtains the secret key that he/she uses to decrypt messages.
(3) Encryption: When patient wishes to encrypt a message, he encrypts the message by computing or obtaining the public key, and then encrypting a plaintext message M with public key to obtain ciphertext C.
(4) Decryption: When doctor has C and private key, he/she decrypts C to obtain the plaintext message M. Here we are providing the application of Identity Based Encryption on ECC and ECIES.

ELLIPTIC CURVE CRYPTOGRAPHY

To setup ECC, we first select a particular elliptic curve E over GF (p), where p is a big prime number. We also denote P as the base point of E and q as the order of P, where q is also a big prime. We then pick a secret key x, and the corresponding public key y, where $y = x \cdot P$, and a cryptographic hash function h(). Finally, we have the secret key x and public parameters (y, P, p, q, h(.)). Encrypting a message m using public key y as EccEncrypt(m, y). The resulting ciphertext is denoted by c. The decryption of ciphertext c using the secret key x is given as EccDecrypt(c, x).

The algorithms for EccEncrypt and EccDecrypt are found in following Alg. 1 and Alg. 2 respectively.

Algorithm1: EccEncrypt(m, y)
1: Generate a random number $r \in GF(p)$. Encrypt m with r, $E_r(m)$ 2: Calculate $A_r = h(r) \cdot y$ 3: Calculate $B_r = h(r) \cdot P$ 4: Calculate $\alpha_r = r \oplus x(A_r)$, where $x(A_r)$ is the x coordinate of A_r 5: Return ciphertext $c = < \alpha_r, B_r, E_r(m) >$

Algorithm2: EccDecrypt(c, x)
1: Calculate $x \cdot B_r = x \cdot h(r) \cdot P = h(r) \cdot y = A_r$ 2: Determine the x coordinate, $x(A_r)$ 3: Derive symmetric key r with $\alpha_r \oplus x(A_r) = r \oplus x(A_r) \oplus x(A_r) = r$ 4: Apply r to $E_r(m)$ to return m

ELLIPTIC CURVE INTEGRATED ENCRYPTION SCHEME

In 1997, Mihir Bellare and Philip Rogaway [28] presented the Discrete Logarithm Augmented Encryption Scheme (DLAES), which was subsequently improved by the same authors and Michel Abdalla, being first renamed as the Diffie-Hellman Augmented Encryption Scheme (DHAES) in 1998 [26] and later as the Diffie-Hellman Integrated Encryption Scheme (DHIES) in 2001 [27], in order to avoid confusions with the Advanced Encryption Standard (AES). DHIES represents an enhanced version of the ElGamal encryption scheme, using elliptic curves in an integrated scheme which includes public key operations, symmetric encryption algorithms, MAC codes, and hash computations. Because of the integration of different functions, DHIES is secure against chosen ciphertext attacks without having to increase the number of operations or the key length [27].

ECIES FUNCTIONAL COMPONENENT

As it name indicates, ECIES is an integrated encryption scheme that uses the following functions:

- Key Agreement (KA): Function used by two parties for the creation of a shared secret.

- Key Derivation (KDF): Mechanism that produces a set of keys from keying material and some optional parameters.
- Hash (HASH): Digest function.
- Encryption (ENC): Symmetric encryption algorithm.
- Message Authentication Code (MAC): Information used to authenticate a message.

The elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based encryption algorithm. It is called integrated, since it is a hybrid scheme that uses a public –key system to transport a session key for use by a symmetric cipher.

ECIES is a public-key encryption algorithm where there is assumed to be a set of domain parameters (K,E,q,h,G). With these parameters, we also add a choice of symmetric encryption/decryption functions which we shall denote Ek(m) and Dk(c). The use of a symmetric encryption function makes it easy to encrypt long messages. In addition instead of a simple hash function we require two special types of hash function:
A message authentication code MAC k(c).

MAC : {0.1}^n * {0,1}^* → {0,1}^m

This acts precisely like a standard hash function except that it has a secret key passed to it as well as a message to be hashed.
A key derivation function KD(T,l)

KD : E * N → {0,1}^*

A key derivation function acts precisely like a hash function except that output length could be quite large. The output is used as a key to encrypt a message hence if the key is to be used in a xor-based encryption algorithm the output needs to be as long as the message being encrypted.

The x-or based encryption requires key derivation and the MAC function to encrypt the message on the basis of x-or operation on bits.

The ECIES scheme works like a one-pass Diffie Hellman key transport, where one of the parties is using a fixed long term rather than an ephemeral one. This is followed by symmetric encryption of the actual message. For example the combined length of the required MAC key and the required key for the symmetric encryption is given by l. The recipient is assumed to have a long-term public /private key pair (Y,x) where

$$Y = [x] G$$

ECIES Encryption
INPUT: Message m and public key OUTPUT: The ciphertext (U,c,r) 1. Choose k € R(1…….q-1) 2. U ← [k]G 3. T ← [k]Y 4. (k1‖k2) ← KD(T,l) 5. Encrypt the message c ← Ek1(m) 6. Compute the MAC on the ciphertext r ← MACk2(c) 7. Output (U,c,r)

ECIES encryption requires a message and the key to encrypt the message.First a random prime number is selected between range.then U is predicted which is public key.
Each element of the ciphertext (U,c,r) is important:
U is needed to agree the ephemeral Diffie Hellman key T.
c is actual encryption of the message.
r is used to avoid adaptive chosen ciphertext attacks.

Notice that the data item U can be compressed to reduce bandwisth, since it is an elliptic curve point.
Notice that the T computed in the decryption algorithm is the same as the T computed in the encryption algorithm since
Tdecryption = [r]U = [x][k]G = [k]([x]G) = [k]Y=Tencryption.

ECIES Decryption
INPUT: Ciphertext (U,c,r) and a private key r. OUTPUT: The message m or an ‘invalid ciphetext’ message. 1. T ← [x]U 2. (k1‖k2) ← KD(T,l) 3. Decrypt the message m ← Dk(c). 4. if r ≠ MACk2(c) then output ‘Invalid Ciphertext’ 5. output m.

BACKGROUND

The Elliptic curve with identity based encryption technique provides the security of single data over the network, but here we are comparing the Elliptic curve integrated encryption scheme using IBE and analyse the comparison between the two elliptic curve standards to secure patient’s record. The elliptic curve using Identity based encryption is the technique where the data reading from the sensors can be encrypted using n number of identities, one for each record of the patient. The doctor can only decrypt a particular data even if he knows the identity of the other records also. The

elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based encryption algorithm. It is called integrated, since it is a hybrid scheme that uses a public –key system to transport a session key for use by a symmetric cipher. The algorithm provides an x-or based encryption technique over key distribution function and the MAC code. The idea is to use multiple sensors in multiple patients, where the sensors notes different reading and collect over to the storage site. The ECIES has an advantage over ECC as our algorithm is very useful for the integration of more than one data at the storage site and to encrypt and decrypt them easily. Also our algorithm used a MAC function to generate number in a more secure way and also this algorithm uses the concept of symmetric as well as asymmetric key.

A body sensor network (BSN), is a network of sensors de-
ployed on a person's body, usually for health care monitor-ing. Since the sensors collect personal medical data, secu-rity and privacy are important components in a body sensor network. At the same time, the collected data has to read- ily available in the event of an emergency.

RESULTS COMPASION BETWEEN ECC & ECIES

Public key: The key generated when the sensors starts reading from the patient.

Signature: The sensors when reads the data and encrypted that data using signatures (verification of the sender and the receiver so that the unauthorised user can't access the data) so that data can't be access eavesdropped and the signatures when matched can be decrypted.

Encrypted Data: The data which is not in actual form but can be converted into another form such that the even if the data is accessed can't understand by the others.

Decrypted Data: The data which is encrypted to provide a security to the data will be decrypted by the same technique used for encryption such that data is correct and readable.

Data storage: The memory required to store a single data from the patient in the sensor.

Parameters	ECC-IBE	ECIES
Public key	0.74 sec	0.69 sec
Signature time	0.77 sec	0.7 sec
Time to encrypt	5.7 sec	5.5 sec
Time to decrypt	1.12 sec	2.07 sec
Storage	1.6 KB	45 bytes

Fig. 1.1 Analysis on different parameters

As shown in the fig. 1 that the proposed algorithm when implemented gives less time than the existing ECC technique, Also the proposed algorithm requires less storage in the sensor to store the data. Hence requires less storage.

Storage (bytes)	ECC-IBE	ECIES
2500	50	N*50
5000	100	N*100
10000	200	N*200
25000	500	N*500
50000	1000	N*1000

Figure 1.2 Key Required Vs Storage Size in Byte

Where 'N' is the number of patients.

As our proposed algorithm generates 'n' number of public keys and different on the number of public keys the sensors read that number of data. As shown in the fig. 1.2 that as the memory required to store the data will depends on the number of public keys and the storage will increase if the number of public keys generated will increase.

Time(S)	3.45	6.9	10.35	13.8	17.25	20.7
Keys	5	10	15	20	25	30

Figure 1.3 Time Required Vs Key Required

Result Analysis

The fig. 1.3 shows the time required to generate the public key, as our proposed algorithm generates 'n' number of public keys, so the time required for generating 'n' public keys will increase the time according to the number of public keys generated.

Conclusion

This paper has presented the working of a system of compact, wearable, wireless body sensing devices implanted in the human body. The novel achievement is that we have proposed is the improvement in the existing protocol for data encryption, decryption and transfer between BSN, storage site and doctor with the need for high data rates.

The main idea to implement the elliptic curve algorithms that may be used for securing of multiple patient's data over sensor network, here we are implementing and analysing two elliptic curve technique and comparing on different parameters.

REFERENCES

[1] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo. Applicability of identity-based cryptography for disruption-tolerant networking. In MobiOpp 2007.

[2] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In CRYPTO 2001.

[3] S. Capkun, L. Butty´an, and J.-P. Hubaux. Self organized public-key management for mobile ad hoc networks. IEEE TMC 2003.

[4] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In IEEE SP 2003.

[5] C. Cocks. An identity based encryption scheme based on. quadratic residues. In LNCS 2260 (2001).

[6] W. Du, R. Wang, and P. Ning. An efficient scheme for authenticating public keys in sensor networks. In MobiHoc 2005.

[7] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In CCS 2002.

[8] R. Ganti, P. Jayachandran, and T. Abdelzaher. Satire: A software architecture for smart attire. In Mobisys 2006.

[9] J. Girao, D. Westhoff, E. Mykletun, and T. Araki.Tinypeds: Tiny persistent encrypted data storage in asynchronous wireless sensor networks. Ad Hoc Networks 2007.

[10] U. Hengartner and P. Steenkiste. Exploiting hierarchical identity-based encryption for access control to pervasive computing information. In SecureComm 2005.

[11] C. Karlof, N. Sastry, and D. Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In SenSys 2004.

[12] A. Kate, G. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In SecureComm 2007.

[13] L. Lazos and R. Poovendran. Serloc: Secure range independent localization for wireless sensor networks. ACM TOSN 2005.

[14] A. Liu, P. Kampanakis, and P. Ning. Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3). 2007.

[15] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In CCS 2003.

[16] B. Lo and G. Z. Yang. Key technical challenges and current implementations of body sensor networks. In BSN 2005.

[17] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In BSN 2004.

[18] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In SECON 2004.

[19] K. Malasri and L. Wang. Addressing security in medical sensor networks. In HealthNet 2007.

[20] M. Mont, P. Bramhall, and K. Harrison. A flexible role-based secure messaging service: exploiting IBE technology for privacy in health care. In International Workshop on Database and Expert Systems Applications 2003.

[21] E. Mykletun, J. Girao, and D. Westhoff. Public key based cryptoschemes for data concealment in wireless sensor networks. In ICC2006.

[22] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In Mobicom 2001.

[23] A. Shamir. Identity-based cryptosystems and signature schemes. In CRYPTO 1984.

[24] L. Zhong, M. Sinclair, and R. Bittner. A phone centered body sensor network platform: cost, energy efficiency and user interface. In BSN 2006.

[25] http://en.wikipedia.org/wiki/Identity_based_encryption.

[26] M. Abdalla, M. Bellare, P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem, contribution to IEEE P1363a, 1998.

[27] M. Abdalla, M. Bellare, P. Rogaway. The Oracle Diffie- Hellman Assumptions and an Analysis of DHIES, Lecture Notes in Computer Science, 2020, pp. 143–158, 2001.

[28] M. Bellare and P. Rogaway. Minimizing the Use of Random Oracles in Authenticated Encryption Schemes, Lecture Notes in Computer Science, 1334, pp. 1–16, 1997.