ISSN:0975-9646
Onkar Kasarlewar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3820-3824

Secure Broker-less Publish/Subscribe System using ECC

Onkar Kasarlewar[#1], Prof. P. S. Desai[#2]

[#]*Department of Computer , Savitribai Phule University of Pune*
Smt. Kashibai Navale College of Engineering, Pune, India

***Abstract*— In case of distributed system, there are two convincing paradigm I) RBAC (Role Based Access Control) and II) Publisher-Subscriber System. Publisher-subscriber system has many advantages over RBAC such as many to many communication, loose coupling between publisher (producer) and subscriber (consumer), and asynchronous in nature. In existing system, AES (Advanced Encryption Standard) is used for an encryption and decryption of an content and bilinear map is used for an key generation used by AES algorithm .In purposed system, ECC is an approach to public key cryptography based on the algebraic structure of elliptic curve over finite values. One of the main benefits of ECC over non-ECC cryptography is the same level of security provided by keys of smaller size. Along with security, it takes low memory usage, CPU utilization and encryption time as well. For handshaking purpose, subscription based tree is used which takes less time than attribute based tree.**

Keywords*— ECC, AES, Broker-Less, Bilinear map, Credentials Publish/Subscribe, Security.

I. INTRODUCTION

Distributed systems are largely being used. In this paper, for study and enhancement in system [1] is taken as base, where the system takes the information provided by various users, said as user credentials, is used to improve security. As traditional point to point communication mechanisms making system more complex and difficult to understand, there is way opened up for loosely coupled communication system. Due to the decoupling of publishers from subscribers, the publish/subscribe (pub/sub) communication model has achieved high popularity. Publishers distribute/publish data into the pub/sub network, and subscribers provide subscriptions which describes their interest in events. Without publishers knowing the set of subscribers, published events are delivered to their appropriate subscribers. This decoupling is generally guaranteed by routing over a broker system [2].

In later systems, publishers and subscribers arrange themselves in a broker less routing framework, building an event forwarding overlay [3].

Content based pub/sub is the model that gives the most expressive subscription framework, where subscriptions provide limitations on the message content. It is helpful for large scale distributed applications because of its expressiveness and asynchronous nature, for example, environmental monitoring, traffic control, and news distribution. Of course, pub/sub needs to give supportive components to complete the essential security requirements of these applications, for example, confidentiality and access control.

In the setting of pub/sub system, access control implies that only authenticated publishers are permitted to distribute events in the system and only those events are delivered to authorized subscribers. In addition, the content of events ought not to be revealed to the routing framework and a subscriber ought to get all related events without exposing its subscription to the system. Solving these security issues in a publish/subscribe framework forces new difficulties. For example, end-to-end authentication utilizing a Public Key Infrastructure clashes with the loose coupling of publishers and subscribers, a key prerequisite for building scalable publish/subscribe system. For PKI, publishers must keep up the public keys of all intrigued subscribers to encrypt events. Subscribers must know the public keys of all related publishers to confirm the authenticity of the events.

Besides, conventional systems to give confidentiality by encrypting the entire event message clash with the content based routing standard. Henceforth, new methodologies are required to route encrypted events to subscribers without knowing their subscriptions and to permit subscribers and publishers authenticate one another without knowing one another. In the existing system, another methodology is provided to enable authentication and confidentiality in a broker less publish/subscribe system. This methodology permits subscribers to maintain credentials as per their subscriptions. Private keys allocated to the subscribers are labeled with the credentials. A publisher assigns each encrypted event with a set of credentials. Also, Identity Based Encryption (IBE) [4], [5] methodologies are implemented to guarantee that a specific subscriber can decrypt an event if there is a match between the credentials labeled with the event and the key; and to permit subscribers to verify the authenticity of events.

II. LITERATURE SURVEY

The literature survey is divided into two categories which are: Approaches based on Broker Network, Approaches based on Semi-Trusted Broker Network.

A. Approaches based on Broker Network:

In the study by C. Raiciu and D.S. Rosenblum [6], they had introduced an investigation of confidentiality in content-based publish/subscribe, attaining some of the security concerns specific to this interaction model. They had displayed a formal security model and analyzed the general C-CBPS issue, indicating out its limitations. They

had provided provably secure procedures that permit content based routing for the huge amount of applications. They have depicted two conventions that support range matches in C-CBPS yet can likewise be applied in different areas.

J. Bacon et al [7] presented architecture which contains administration domains sharing a dedicated event-broker system. They had discovered this to be suitable for some applications. They likewise accepted a secure server per domain that handles credentials and activates parts as per policy. With access control functionality located in the client-hosting brokers, they had the capacity to enforce RBAC on the publish/subscribe clients. Generally, separating event-management functionality into event service makes access control simpler to enforce than in a peer-to-peer methodology where the client and event service are co located. The latter appears unsuitable for applications transmitting sensitive data. They have expected content based routing, for proficiency of communication, instead of broadcast routing. At the point when a few brokers are not trusted to see particular sensitive information this style of routing can be utilized, with the modifications they provided.

M. Ion et al [8] given a solution for providing confidentiality in pub/sub frameworks. Their solution is an encryption plan based on CP-ABE, KP-ABE and multi-client SDE. Their plan supports both the publication and the subscription confidentiality property while in the meantime does not oblige publishers and subscribers to share secret keys. In spite of the fact that events and filters are encrypted, brokers can perform event filtering without realizing any data. At last, their plan permits subscribers of express filters that can characterize any monotonic and non-monotonic imperatives on events.

M. Srivatsa et al [9] have introduced EventGuard, reliable framework for ensuring publish/subscribe services from different attacks. EventGuard offers security features that are basic to publish-subscribe overlay services, for example, confidentiality, authenticity, integrity, and strength to flooding based DoS attacks. We have depicted the two key segments of EventGuard. The first segment is a suite of security guards that protects the essential publish and subscribe operations from DoS attacks and unauthorized reads and writes. The second segment is a flexible publish-subscribe system design that is equipped for giving secure yet adaptable message routing, countering message dropping-based DoS attacks. A remarkable feature of EventGuard is its combined security structure that meets both security objectives for shielding the pub-sub overlay services from different susceptibility and threats and performance objectives for keeping up the adaptability and simplicity of the general framework while giving security guarantees.

S. Choi et al [10] presented a safe CBPS framework based on Asymmetric Scalar product Preserving Encryption to offer notification and subscription confidentiality and to diminish matching complexity. Their techniques help range filtering, equality filtering, covering, conjunction filtering, and inequality filtering, which are crucial in CBPS. Furthermore, their solution does not cause false positives, rather than existing work, for example, C-CBPS. Also, they proposed another technique for secure aggregation utilizing homomorphic functions and ASPE.

B. Approaches based on Semi-Trusted Broker Network:

Objective of P. Pietzuch [11] was to develop Hermes event-based middleware platform. They depicted its layered architecture and the two event routing algorithm supported by Hermes, type based routing, which supports subscriptions as per an event type, type-and-attribute-based routing, which gives content-based filtering with respect to event attributes also. Both routing algorithms utilize meeting nodes to develop adaptable event dissemination trees on top of a distributed hash table. Due to their prerequisite of programming language integration, they developed the routing algorithms with event type inheritance and support for supertype subscriptions. They likewise presented the fault-tolerance mechanisms in the algorithms that are focused around soft-state approach and the replication of meeting nodes. A model implementation of Hermes was proposed in detail, as was an assessment of Hermes routing in a distributed frameworks simulator, contrasting it with the Siena routing algorithm, which is standard for content-based routing of events.

L. Opyrchal and A. Prakash [12] recognized the "safe end-point delivery" issue and investigated various probable solutions. They were concerned about providing confidentiality when sending events from brokers to subscribers. The issue is that in content-based frameworks, each event can possibly have a different set of intrigued subscribers. There are $2N$ probable subsets, where N is the number of subscribers. With a huge number of subscribers it is infeasible to setup static security groups for each probable subset. Various key management frameworks for group communication tackle a similar issue yet none of them was intended to handle the dynamic nature of content based event delivery. They investigated various dynamic caching methodologies. A basic solution is to encrypt every event independently for each one interested subscriber; however this obliges a huge number of encryptions for substantial sets of subscribers. Their primary objective is to decrease the number of encryptions needed to protect privacy while sending events to intrigued subscribers. The number of encryptions is essential as it makes an interpretation straightforwardly into message throughput.

H. Khurana [13] demonstrated a solution for providing confidentiality, integrity, and authentication of events as they are routed through a content based publish/subscribe network. Their solution supposes an untrusted broker network. In this context their solution allows brokers to do content based matching and routing with respect to cleartext parts of events yet does not reveal sensitive event content to the brokers as they are encrypted. The solution utilizes Jakobsson's proxy re-encryption methods to disseminate event encryption keys by means of a transformation methodology to authorized subscribers without obliging any direct interactions of publishers and subscribers. Additionally, they provide verifiable usage-based accounting services by logging all transformations and

giving publishers with the logs so they can charge subscribers.
Algorithms:
I.AES:
1: Key Expansion: - Using Rijndael's key schedule Round keys are derived from the cipher key.
2: If DistanceToTree(u) > DistanceToTree(DCM) and First-Sending(u) then
3: Initial Round :- AddRoundKey where Each byte of the state is combined with the round key using bitwise xor.
4: Rounds

- SubBytes : non-linear substitution step
- ShiftRows : transposition step
- MixColumns : mixing operation of each column.
- AddRoundKey

5: Final Round: It contain SubBytes, ShiftRows and AddRoundKey

Decryption of the cipher texts follows reverse of encryption steps. These rounds repeat for 10/12/14 times for a particular data depends on key size 128/192/256.
II.ECC
In case of ECC, If two communicating parties want to Communicate the messages, they agree upon to use an elliptic curve Ep (a,b) where P is a prime number and a random point C on the elliptic curve. The standard equation for an elliptical curve is y2=x3+ax+b, where a and b are coefficient for that equation.
1. Using standard equation, make code table using different values of x and y.
2. Sender selects 'b' less than 'P 'and any point 'B' on curve whereas receiver selects 'a' less than 'P' and any point 'A' on curve. These points are private keys of senders and receivers.
3. A1=a*(C+A), A2=a*A where A1,A2 are general public keys of receiver and B1=b*(C+B) and B2= b*B where B1 and B2 are general public keys of sender.
4. Sender and receiver make special keys for each other. Sender calculates S=b*A2 and receiver calculates P=a*B2.
5 .If senders want to encrypt data, then data encrypted character by character.
Each character encrypted by,
E1=x*C, Where c is random number
E2=Plain text +(b+x)*A1-(x*A2)+P
6. Receiver decrypt the data using
M=E2-(a*E1+a*B1+B2)

III. PROPOSED SYSTEM

PHASE-I:
In the proposed system, both publisher and subscriber are allowed to maintain their confidentiality and produces key which is used for a handshaking between them. Key server collects handshaking keys from both publishers and subscriber and as per the subscricption of subscriber,it changes xml file (xml tree) eg
.<subscriber_name>
<todate>
<public keys of publisher event>
.
.
<public keys of publisher events>
<public keys of publisher events>
<fromdate>
</subscriber_name>

PHASE II:
Publishers publish an events and subscribers get an content and abstract of an events he subscribed.ECC produces public and private keys and publisher sends encrypted data to the subscriber and subscriber decrypt the data with private keys. If subscribers wish to change the particular data of the publisher, he request publisher to get write right. Publisher could give grant to write data or decline. If publisher allow subscriber to change the data , subscriber can change the data and that changed data send to the publisher again for final permission.If publisher allows the change data, then that particular data will change in data base.

System Architecture

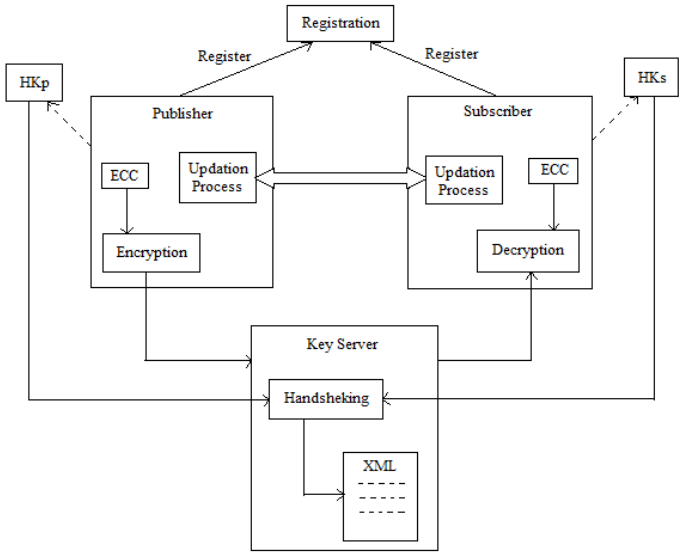


Fig. 1. System ArchitectureThis architecture of the system shows complete flow of data between system components.

A. *Mathematical Model*

Set Theory
System S is represented as S= {R, C, CT,K,EC,E,D}

1) *Registration Process*

R= {P, S}
Where, R is the set of publishers and subscribers
i. P= {p1, p2, p3,…..pn}
Where, P is represented as a set of publishers and p1, p2…pn are the number of publishers.
ii. S={s1, s2,….sn}
Where, S is the set of subscribers and s1, s2….sn are the number of subscribers.

2) *Credentials*

C= {N, CT,D1,D2}
Where C is the set of credentials, N is name of publisher or subscriber, CT is category in which publisher and subscriber belongs.D1 is date of publication and D2 is date of subscription.

3) Categories for Events

CT= {SC, R, SP, P, E, W}

Where, CT is set of Categories and SC is for sports, R is for religion, SP is for sports, P is for politics. E is for entertainment and W is for weather.

4) Keys

K = {HK, PK, SK}

Where, K is set of keys in system

HK= {HK_P, HKS}

Where, HK is set of keys for handshaking between publishers and subscribers.

HK_P= {HK_{P1}, HK_{P2},…..}

Where HK_P represents key for handshake in case of publishers.

HK_S= {HK_{S1}, HK_{S2},…}

Where HK_S represents key for handshake in case of publishers.

PK={PK_P,PK_S}

Where, PK represents set of public keys for subscriber and publishers, PK_P represents public key for a publisher, PK_S represents public key for subscriber.

SK={SK_P,SK_S}

Where, PK represents set of public keys for subscriber and publishers, PK_P represents public key for a publisher, PK_S represents public key for subscriber.

5) Elliptic Curve

EC represents $E_P(a,b)$ which is an Equation for an elliptic curve with coefficient a and c , and P represents random prime number.

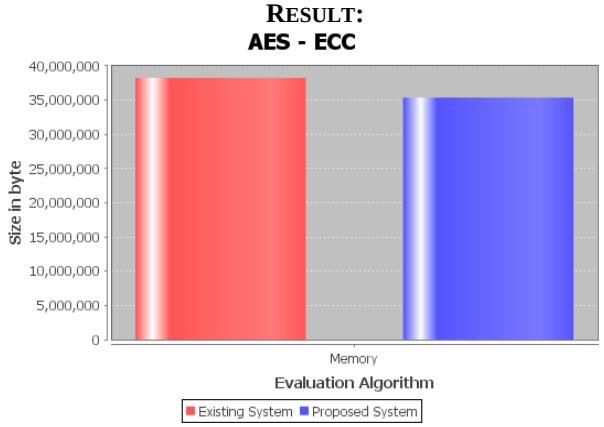
6) For Encryption and Decryption

E = ECC (plain text, PK_S) → cipher text

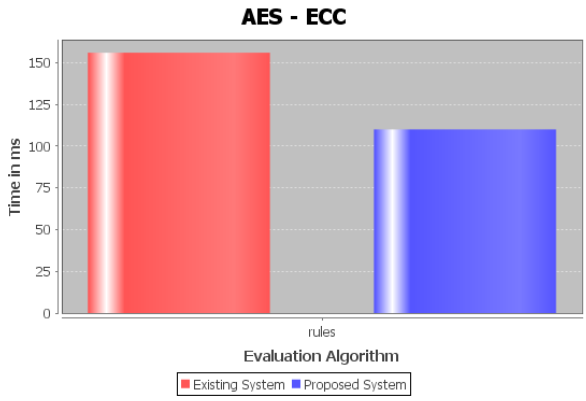
D = ECC (cipher text, SK_S) → plain text

Where, ECC is Elliptic curve cryptography alternative to public key encryption.

RESULT:



The above graph based on the Memory Graph X-axis shows the Evaluation Algorithm. In this figure again two bars are shown first is of Existing System and second one is Proposed System. On Y-axis it shows on Size in bytes where it starts from 0 to 25,000,000 bytes. In this graph Existing System is takes almost to 25,000,000 bytes to perform operation, and the Proposed System performs operations quicker.



The above graph depicts the comparison between AES and ECC algorithm where X-axis shows the Evaluation Algorithm. In this figure two bars are shown first is of Existing System and second one is Proposed System. On Y-axis it shows on time in ms where it starts from 0 to 600ms. In this graph Existing System is takes up to 600 ms to get execute, Proposed System executes quicker.

CONCLUSION

The methodology proposed here provides same level of security provided by keys of smaller size. Credentials used for the handshaking between authorized publishers and subscribers. The Elliptic curve cryptography (ECC) uses less computational power, communication bandwidth, and memory as compare to other cryptosystems.AES takes more time and memory as compare to the ECC. So proposed work is more effective in case of many devices like wireless communication devices, embedded systems which has limited storage and computational power.An application where security is needed but lacks the power, storage and computational power ,our purpose work is useful.

REFERENCES

[1] M. Tariq, B. Koldehofe, and K.Rothermel, “Securing Broker-Less Publish/Subscribe Systems Using Identity-Based Encryption,” IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014

[2] H.-A. Jacobsen, A.K.Y. Cheung, G. Li, B. Maniymaran, V. Muthusamy, and R.S. Kazemzadeh, “The PADRES Publish/Subscribe System,” Principles and Applications of Distributed Event-Based Systems. IGI Global, 2010.

[3] M.A. Tariq, B. Koldehofe, G.G. Koch, I. Khan, and K. Rothermel, “Meeting Subscriber-Defined QoS Constraints in Publish/Subscribe Systems,” Concurrency and Computation: Practice and Experience, vol. 23, pp. 2140-2153, 2011.

[4] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” Proc. IEEE Symp. Security and Privacy, 2007.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” Proc. ACM 13th Conf. Computer and Comm. Security (CCS), 2006.

[6] C. Raiciu and D.S. Rosenblum, “Enabling Confidentiality in Content-Based Publish/Subscribe Infrastructures,” Proc. IEEE Second CreatNet Int’l Conf. Security and Privacy in Comm. Networks (SecureComm), 2006.

[7] J. Bacon, D.M. Eyers, J. Singh, and P.R. Pietzuch, “Access Control in Publish/Subscribe Systems,” Proc. Second ACM Int’l Conf. Distributed Event-Based Systems (DEBS), 2008.

[8] M. Ion, G. Russello, and B. Crispo, “Supporting Publication and Subscription Confidentiality in Pub/Sub Networks,” Proc. Sixth Int’l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), 2010.

[9] M. Srivatsa, L. Liu, and A. Iyengar, "EventGuard: A System Architecture for Securing Publish-Subscribe Networks," ACM Trans. Computer Systems, vol. 29, article 10, 2011.

[10] S. Choi, G. Ghinita, and E. Bertino, "A Privacy-Enhancing Content-Based Publish/Subscribe System Using Scalar Product Preserving Transformations," Proc. 21st Int'l Conf. Database and Expert Systems Applications: Part I, 2010.

[11] P. Pietzuch, "Hermes: A Scalable Event-Based Middleware," PhD dissertation, Univ. of Cambridge, Feb. 2004.

[12] L. Opyrchal and A. Prakash, "Secure Distribution of Events in Content-Based Publish Subscribe Systems," Proc. 10th Conf. USENIX Security Symp., 2001.

[13] H. Khurana, "Scalable Security and Accounting Services for Content-Based Publish/Subscribe Systems," Proc. ACM Symp. Applied Computing, 2005.