Attribute Based Sharing in Cybersecurity Information Exchange Framework

Iman Vakilinia  
Dept. of Computer Science and Eng.  
University of Nevada, Reno, NV, USA  
ivakilinia@unr.edu

Deepak K. Tosh  
Dept. of Computer Science  
Norfolk State University, VA, USA  
dktosh@nsu.edu

Shamik Sengupta  
Dept. of Computer Science and Eng.  
University of Nevada, Reno, NV, USA  
ssengupta@unr.edu

Abstract—As the complexity of the cyber attacks are increasing, there is a growing demand for proactive defense against them. CYBersecurity information EXchange (CYBEX) is playing a crucial role to implement proactive defense. CYBEX conveys organizations’ sensitive information which demands proper access control management. However, previous works in this area do not consider access control for CYBEX. In this work, we tackle the access control problem in CYBEX. We model the attribute based access control in CYBEX with a semi-trusted sharing server. To achieve attribute based access control in CYBEX, our mechanism is developed based on the concepts of ciphertext policy attribute based encryption (CP-ABE) [1] and STIX [2]. The mechanism’s workflow is as follows, at the beginning users claim their attributes from attribute authorities, then a key generation center generates decryption keys for users based on their attributes. For the sharing, organizations embed access control to their shared data. This is conducted by encrypting sensitive information such that only users with appropriate attributes can decrypt them. Security analysis, implementation and performance evaluation indicate the effectiveness and efficiency of the mechanism.

Index Terms—access control, cybersecurity information sharing, attribute based encryption, STIX, CybOX

I. INTRODUCTION

Frequent cyber attacks on private/government organizations and cyberspace consumers have triggered a strong alertness among cybersecurity professionals and hint at the necessity of proactive cyber-defense mechanisms. Although maliciousness around any organization may not be eradicated completely with sole security research, the impact of such crimes can be reduced via enabling collaborative exchange of cyber-threat related information among peer organizations [3]. With such knowledge sharing, it is possible to enhance the cybersecurity awareness of the organizations and potentially provide timely recommendations to safeguard their critical assets from future cyber-exploitations at a reduced cost. Such effort has been fostered through the approval of the recent bill on “Cybersecurity Information Sharing Act (CISA)” [4] from U.S. Congress.

The CYBersecurity information EXchange (CYBEX) [5] aims to automate the cyber-threat intelligence (CTI) sharing process by provisioning various services such as incident representation, exchange protocol, discovery, identity assurance, query services, etc. To facilitate CTI sharing, it is required to have a formal and standard structured language for representing threat information. The DHS-led widely adopted technical specification, Structured Threat Indicator eXpression (STIX) [2], fulfills this requirement and intends to capture the diverse knowledge set on any cyber-incident. STIX provides flexibility of documenting various threat related information in form of an XML, where the attributes refer to a specific block of information related to a cyber incident. STIX directly leverages Cyber Observable eXpression (CybOX) [2] for its representations of observables. DHS has also put forward a standard information transport specification named Trusted Automated Exchange of Indicator Information (TAXII) [6] to share STIX documents among corporations.

Despite all these efforts and understanding the benefits of such exchanges, the organizations are still reluctant to participate in CTI sharing due to several reasons: (1) the possibility of personally identifiable information leaked through the shared report, (2) risk of further exploitation, (3) chances of reputation loss etc. Therefore, it is of a high requirement to preserve the privacy of organizations during the information dissemination process, so that opportunities of cyber-exploitations from the shared information can be inhibited. In reality, rational organizations may not completely cooperate with others in the sharing framework because of market competition and trust relationships among each other. Thus, while sharing threat information, organizations would prefer to enable access control over the shared information so that only the intended entities will be able to successfully access the specific block of information instead of the complete document. Addressing this particular challenge will help the organizations to control their sharing depending on their preferences over other participants and criticality nature of information they choose to exchange. Additionally, enforcement of access control will also hinder malicious agents to exploit the organizations’ sensitive information. Although participants can sign non-disclosure agreements (NDA), but NDAs are not systematic nor scalable. To provision such service, US-CERT has developed Traffic Light Protocol (TLP) [3], which defines a set of designations (in form of colors: red, amber, green, and white), signifying the degrees of information sensitivity and the corresponding sharing considerations. However, it is not applicable with a semi-trusted sharing server and also it is not possible to accomplish fine-grained access control. Thus these methods carry the risk toward security and privacy.

In this work, we tackle the access control problem in the CYBEX. We model the attribute based access control in the CYBEX with a semi-trusted sharing server. The proposed framework involves multiple entities: (1) Users who access
the sharing CYBEX information, (2) Organizations which share CYBEX information, (3) Attribute authorities which are responsible to assign credentials to the users, (4) Key generation center which is responsible to setup the system, and (5) Server which is responsible to store information. To achieve attribute based access control in the CYBEX, our mechanism is developed based on ciphertext policy attribute based encryption (CP-ABE) [1] and STIX [2]. At the beginning users claim their attributes from attribute authorities. Key generation center provides decryption keys for users based on their attributes. Organizations embed access control to their shared data. In order to do that they encrypt their sensitive information such that only users with appropriate attributes can decrypt them.

The rest of the paper is organized as follows. Next section reviews major works in cybersecurity information exchange and access control. In Section III and IV, we introduce our system model and building blocks of our mechanism, respectively. Details of our proposed mechanism are described in Section V. In section VI, implementation of the proposed mechanism has been discussed. We analyze security of our mechanism in Section VII. Section VIII is the performance evaluation. We conclude our paper in Section IX.

II. RELATED WORK
A. Cybersecurity Information Exchange

The bill “S.754-Cybersecurity Information Sharing Act (CISA) of 2015” [7] has been passed by the U.S. senate. The bill encourages private companies, businesses, and federal organizations to share cyber-threat information with one another. To facilitate such sharing framework, ITU-T (International Telecommunication Union-Telecommunication) took the initiative to adopt the Cybersecurity Information Exchange (CYBEX) [5] to tighten cybersecurity and infrastructure protection. Various protocols and specifications for cybersecurity information sharing such as TAXII, STIX, CybOX, VERIS, MAEC, SCAP, IODEF [6], [8]–[10] have also been developed. Several information sharing programs, such as CISSP, NCCIC, ISAC, ISAO [11] are also implemented by the Department of Homeland Security (DHS) to provide a collaborative platform for understanding cybersecurity risks and defenses. Although these works aim to provide a platform for information sharing among organizations, but they do not consider protection of sensitive information inside CYBEX messages. Since CYBEX messages might convey sensitive and private information, access control management for the shared messages should be under the control of the information originator organizations. With the presence of big-data [12] and the problem of social group formation [13], it is harder to address the access control challenge. In order to tackle the privacy issues of information sharing, We have presented a privacy-preserving CYBEX mechanism in [14], also we have investigated the best strategy for sharing information in CYBEX in [15]. On the other hand, the game theoretic aspects of CYBEX information sharing has been investigated in [16]–[19]. Unlike the above works, in this work we target to implement a CYBEX system where organizations can manage the access control of their cybersecurity information without having a trusted server.

B. ABE Access Control

There are plenty of works that have been done in the area of access control [20]–[22]. Attribute Based Encryption is a cryptographic solution for access control [23] which is a relatively new approach. This method supports outsourcing data to the untrusted server. In this case, access control is managed by the data owner instead of the storage server. Ciphertext Policy Attribute Based Encryption (CP-ABE) [1] allows the data owner to encrypt the data with consideration of access policy. Users are only able to decrypt the ciphertext if and only if their attributes can pass the access policy of the ciphertext. There are single-authority [1] and multi-authority [24] CP-ABE frameworks. In the single authority setting, the management of attributes and their corresponding keys are handled by one authority, while in the multi-authority CP-ABE systems, multiple authorities manage the attributes. We are utilizing a simple multi-authority CP-ABE system to authorize access to the sensitive cybersecurity messages. Since access control is a crucial parameter of CYBEX and previous works did not consider it, in this work we focus on applying CP-ABE into the CYBEX system. In order to achieve this goal, we present a CYBEX sharing framework. This framework is consisted of three components (i) Registration, (ii) Sharing, and (iii) Access.

III. SYSTEM MODEL

A. Entities and Architecture

In this section, we give an overview of our mechanism, which includes the following three components: (i) Registration, (ii) Sharing, and (iii) Access. We consider the system consists of five entities: (1) a trusted key generation center $KGC$, (2) a set of participating organizations $O_i$, $1 \leq i \leq N$, (3) storage server $S$, (4) attribute authorities $A_j$, $1 \leq j \leq L$, and (5) users accessing the shared information $\mathcal{U}_k$, $1 \leq k \leq M$. $KGC$ is a trusted entity responsible to manage CP-ABE key generation. $S$ provides the hosting service for sharing cybersecurity information. We consider $S$ as an honest-but-curious entity. i.e. $S$ strictly follows the protocol but it is curious about the sensitive data which $O_i$ shares. $A_j$ is responsible for issuing attributes to $\mathcal{U}_k$. Notice that $O_i$, $A_j$ and $\mathcal{U}_k$ can be the same entity. The system should be 	extit{collusion-resistance} which means if multiple $\mathcal{U}$ collude, they should not be able to decrypt any ciphertext which any of them is not able to decrypt solely. We classify the CYBEX messages as sensitive and non-sensitive. The sensitive message contains private information which $O_i$ does not want to reveal to the public (e.g. system configuration, zero-day vulnerability, users private information, and etc), on the other hand the non-sensitive message is shared publicly (e.g. blacklist IP addresses). Our mechanism aims to protect sensitive message and for the rest of the paper, we will focus on it.

A typical workflow of the entire system can be described as follows. First the participant $O_i$ and $\mathcal{U}_k$ register to the CYBEX program. $O_i$ submits the attributes and their corresponding
A's verification public key to KGc. In the sharing phase, Os encrypt the message with consideration of an access policy such that Uks is able to decrypt the ciphertext if and only if it has the required attributes. In order to access to the shared cybersecurity information, Uks receives the decryption key from KGc based on its attributes. In the access phase, Uks first checks if its attributes pass through the access policy of the ciphertext and then decrypts the message. Fig. 1 displays the architecture of the proposed mechanism.

![CTI and CYBEX Server](http://example.com_CT1.png)

Fig. 1. CTI and CYBEX Server

**B. Threat Model**

Here we investigate the main challenges of our mechanism. Specifically we aim to achieve the Access-Control in the CYBEX with a semi-trusted S. This means Os should be able to limit the access to the sensitive information shared through S. The mechanism should be Collusion-Resistance, i.e. the colluding Uks should not be able to share their attributes with each other. In other words, decryption keys which is associated with attributes are not extensible by Uks. There should also be a mechanism in the system which allows Os to revoke the attributes from Uks.

**IV. BUILDING BLOCKS**

Here we describe the preliminary building blocks of our mechanism.

**Bilinear Map.** [25] Let $G_0$ and $G_1$ be cyclic multiplicative groups of prime order $p$, where each group has unique binary representation. $g$ is a generator of $G_0$. The bilinear map $e : G_0 \times G_0 \rightarrow G_1$ has the following properties

1) Bilinearity: $\forall x, y \in G_0$ and $a, b \in \mathbb{Z}_p^*$, $e(x^a, y^b) = e(x, y)^{ab}$.

2) Non-degeneracy: $e(g, g) \neq 1$

We say that $G_0$ is a bilinear group if the group action in $G_0$ and the bilinear mapping $e$ are both efficiently computable.

**Access Tree.** [1] Access tree $\tau$ specifies the access policy that private keys must satisfy to decrypt the ciphertext. Each interior node of $\tau$ is a threshold gate ("AND" or "OR") and the leaves represent the attributes. Number of children of a node $x$ is $n_x$ and $t_x$ is its threshold value. We have $0 < t_x \leq n_x$. OR gate has $t_x = 1$ and AND gate has $t_x = n_x$. The leaf node $x$ is an attribute with a threshold value $t_x = 1$. $p_x$ is the parent node of $x$. $a_x$ denotes the attribute associated with the leaf node. The access tree defines ordering for nodes, the $i_x$ denotes this index for node $x$. $\tau_x$ is the subtree of $\tau$ rooted at the node $x$. If $\tau_x(\gamma) = 1$ then the attribute set $\gamma$ satisfies the access tree $\tau_x$.

**CP-ABE.** [1] In the construction of CP-ABE, the data owner encrypts the message based on the access tree policy. A user is able to decrypt the ciphertext if his attributes satisfy the ciphertext access tree. In this mechanism users receive their decryption keys based on their attributes. CP-ABE contains following algorithms:

**Setup:** This algorithm chooses a bilinear group $G_0$ of prime order $p$ with generator $g$, then it randomly selects $\alpha, \beta \in \mathbb{Z}_p^*$ and the public key is published as:

$$PK = \{G_0, g, h = g^{\beta}, e(g, g)^{\alpha}\}$$

The master key is $(\beta, g^\alpha)$.

**KeyGen(MK, $\gamma$):** This algorithm takes as input the master key $MK$ and attribute set $\gamma$ and outputs the secret key $SK$ which is associated with the input set. It first chooses a random $r \in \mathbb{Z}_n^*$, and then for each attribute $j \in \gamma$ chooses a random $r_j \in \mathbb{Z}_n^*$. Afterward it computes the key as:

$$SK = \{D = g^{(\alpha+\gamma)/\beta}, \forall j \in \gamma : D_j = g^r \cdot H(j)\}$$

**Encrypt:** This algorithm encrypts the message $M$ with public key $PK$ under the access tree structure $\tau$. The algorithm first traverses in the tree $\tau$ and set a polynomial $q_x$ for each node $x$. In order to set polynomials, the algorithm in a top down manner starts from the root node $R$ and chooses polynomials in the following way. For each node $x$ in the tree, the degree $d_x$ of the polynomial $q_x$ is set to be the threshold value $k_x$ of that node minus one, that is $d_x = k_x-1$. The algorithm starts from the root node $R$ and chooses a random $\epsilon \in \mathbb{Z}_n^*$ and sets $q_R(0) = \epsilon$. Then, it randomly chooses $d_R$ number of other points of the polynomial $q_R$ to define the polynomial. For any other node $x$, it sets $q_x(0) = q_R(i_x)$. It again randomly chooses $d_x$ other points to define the polynomial $q_x$. If $Y$ be the set of leaf nodes in $\tau$, then the ciphertext is constructed by giving the access tree structure $\tau$ as input

$$CT = \{\tau, \tilde{C} = Me(g, g)^{\alpha'}, C = h^\epsilon, \forall y \in Y : C_y = g^{\alpha'(0)}, C'_y = H(a_y^{\alpha'(0)})\}$$
Decrypt(CT, SK, p): This algorithm runs DecryptNode(CT, SK, x) for each leaf node \( x \in \tau_p \) with \( i = a_x \).

\[
\text{DecryptNode}(CT, SK, x) = \frac{e(D_i, C_p)}{e(D_i, C_p^\tau)} = \frac{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)\cdot r_i})}{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)\cdot r_i})} = e(g, g)^{q_x(0)}.
\]

Then the algorithm computes \( \phi_p \) where \( p \) is the root of subtree \( \tau_p \), \( \Delta_{i, \gamma}(x) = \prod_{j \in \gamma, j \neq i} \frac{x - j}{i - j} \), \( i = i_z, \gamma'_z = i_z : z \in \gamma_x \).

\[
\phi_p = \prod_{z \in \gamma_x} \phi_z^{\Delta_{i, \gamma'(0)}} = \prod_{z \in \gamma_x} (e(g, g)^{r \cdot q_z(0)})^{\Delta_{i, \gamma'(0)}} = \prod_{z \in \gamma_x} (e(g, g)^{r \cdot q_z(0)})^{\Delta_{i, \gamma'(0)}} = \prod_{z \in \gamma_x} (e(g, g)^{r \cdot q_z(0)})^{\Delta_{i, \gamma'(0)}} = e(g, g)^{r \cdot q_x(0)}.
\]

Then the algorithm sets \( A = e(g, g)^{r \cdot r_e} \) and decrypts CT by computing:

\[
\tilde{C} / (e(C, D)/A) = \tilde{C} / (e(h^s, g^{(\alpha + r) / \beta}) / e(g, g)^{r \cdot r_e}) = M.
\]

V. ATTRIBUTE-BASED SHARING

A. Registration

At the beginning, KGC generates the public values of the system by running the Setup algorithm. Afterward, \( O_i \) registers to the CYBEX program. \( O_i \) submits the attributes and their responsible \( A_j \)'s public keys to KGC. Here there is a trust relationship between \( O_i \) and \( A_j \). Although \( O_i \) and \( A_j \) can be the same entity, this procedure evaluates the scalability of the framework. \( O_i \) exploits the registered attributes as conditions to limit the access to the sensitive information. Then \( U_k \) requests attributes from corresponding \( A_j \). In this step \( U_k \) provides its credentials to \( A_j \). For example, consider that a financial institute receives the "Financial" attribute from the financial-center-authority. After verification of the \( U_k \)'s credentials, \( A_j \) issues a signature \( \sigma_{(CIDU_k, a_i)} \) indicating \( U_k \) has attribute \( a_i \) for the defined CYBEX program \( CID \). Each CYBEX program is time bounded. Attributes are valid in the current CYBEX program and \( U_k \) has to obtain the new CYBEX access attributes for each CYBEX program. This allows \( O_i \) to manage access revocation and \( A_j \) can deny \( U_k \)'s new request for the next CYBEX program.

After gathering signatures for attributes, \( U_k \) sends the set of received signatures \( \Sigma_{CIP} \) to KGC. KGC first verifies the validity of the signatures. If all of the signatures are valid, then KGC generates the decryption keys for \( U_k \) by executing the key generation algorithm KeyGen(MK, \( \gamma \)). At the end of this phase, \( U_k \) has received its secret key \( SK \) from KGC. The whole process is depicted in Fig. 2.

B. Sharing

At the time of sharing cybersecurity information, \( O_i \) first generates the access tree \( \tau \). The access tree defines access structure for the message \( M \). In other words, \( O_i \) defines the authorization rules to access the cybersecurity shared information. Then \( O_i \) executes Decrypt(PK, M, \( \tau \)) algorithm to generates the ciphertext \( C.T \). \( O_i \) embeds \( C.T \) into the STIX message and sends it to the \( S \). \( S \) acts as a storage server and it is not providing any access control service. \( U_k \) has access to all of the information stored in \( S \).

C. Access

In order to access to the sensitive cybersecurity shared information, \( U_k \) first checks the access structure of the ciphertext. If \( U_k \)'s attribute set \( \gamma \) passes through \( \tau \) of \( C.T \) then \( U_k \) first creates subtree \( \tau_p \) and executes Decrypt(CT, SK, p) algorithm to retrieve message.

VI. IMPLEMENTATION

In this section, we investigate the implementation of CP-ABE into the STIX messages. In order to do that, we have extended CybOX object for the representation of CP-ABE rather than creating entirely new CybOX Objects. For instance, consider that the security manager of Bank X wants to share cybersecurity information in the CYBEX program. Since this data conveys sensitive and private information, the security administrator applies the following access structure: "\( X_{Bank\_Branch\_AND\_Security\_Engineer} \) OR "\( Financial\_Institute\_AND\_Location\_USA\_AND\_CISO(Chief\_Information\_Security\_Officer) \)” to encrypt the message. In this case, the sensitive information could only be seen by security engineer agents of Bank X and also CISOs of financial institute located in USA. Fig.3 shows the previous example access tree and Fig.4 displays the corresponding CybOX format with CP-ABE. In this example The Artifact object contains sensitive information. This object is intended to encapsulate and conveys the content of a Raw Artifact. It has also the Encryption attribute. We are setting this attribute to CP-ABE. As described in section V, CP-ABE ciphertext contains five elements \( \{ \tau, C, C', Y \} \). We allocated a special section for displaying \( \tau \), and other parameters reside in the ciphertext section which is ArtifactObj:Raw_Artifact in our example. Sensitive information can reside in different sections of STIX reports. It is possible to only encrypt the specific XML section with sensitive information. At the access phase, \( U_k \) first traverses \( \tau \) to see if its attributes can pass through the access tree, and then exerts the decryption algorithm. \( U_k \) also can investigate the requirements to access the sensitive information. In this case if \( U_k \) finds some sensitive information useful for itself and if it is eligible for \( \tau \) attributes, then it might apply for those attributes in the next period of CYBEX.
VII. SECURITY ANALYSIS

The security of our mechanism directly comes from the ciphertext-policy attribute based encryption [1]. This cryptographic approach allows data owners to perform access control by encrypting messages before they publish them. By this, only the users with sufficient attributes are able to decrypt the messages. CP-ABE also provides the collusion-resistance feature which prevents colluding users to share their attributes in order to gain extra access capability. Access revocation is managed at the end of each CYBEX sharing-program. In this case, $U_k$ should pass new requirements for gaining new attributes from $A_j$. Here, while $S$ provides access to the CYBEX information, it does not have access to $O_i$’s sensitive information.

VIII. PERFORMANCE EVALUATION

In this section, we evaluate the computation performances of the proposed framework. The testbed is built on Linux Ubuntu 15.04 with 2.5GHz CPU and 4GB RAM. We have utilized cpabe toolkit (http://acsc.cs.utexas.edu/cpabe/) for the implementation of CP-ABE in our framework. This toolkit uses PBC library [26] and performs on a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field. In the registration phase, we apply ECDSA digital signature [27] because of its efficiency. We have implemented ECDSA through OpenSSL library (https://www.openssl.org). ECDSA’s elliptic curve is over a prime field of $n = 256$ bits. In the registration phase, $U_k$ receives credentials from $A_j$ and then present them to $KGC$. Credentials are ECDSA signatures. ECDSA signing operation takes 0.528 (ms) and ECDSA verification takes 0.512 (ms). Afterward $KGC$ issues decryption key for $U_k$. Setup public key and master key takes 14 (ms). Key generation algorithm is linear with respect to the number of attributes associated with $U_k$. In the sharing phase, encryption time is linearly dependent to the number of access tree leaf nodes. Finally in the access phase, the decryption is completely related to the access tree. In order to estimate the running time of this phase, we have assigned random values to the access tree and also $U_k$’s available attributes. Fig. 5
Fig. 5. Computation time under different number of attributes displays running time under different number of attributes for Key-generation, Encryption, and Decryption. The result displays that the proposed mechanism is practical even with complex access trees and large decryption keys.

IX. CONCLUSION

In this paper we have proposed a new model for the attribute based access control in CYBEX with a semi-trusted sharing server. This model is based on CP-ABE and STIX. Our model consisted of three phases namely Registration, Sharing, and Access. System setup is conducted in the registration phase. In the sharing phase, organizations encrypt their data based on their intended access policy. Finally users decrypt messages based on their attributes at the access phase. Security analysis, implementation and performance evaluation indicate the effectiveness and efficiency of our mechanism.

X. ACKNOWLEDGEMENT

This research is partially supported by the National Science Foundation (NSF), USA, Award #1528167 and Award #1516724.

REFERENCES