IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 29, 2019, accepted October 27, 2019, date of publication November 12, 2019, date of current version November 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953227

Reliable Access Control for Mobile Cloud Computing (MCC) With Cache-Aware Scheduling

FARA JAMAL[1], MOHD TAUFIK ABDULLAH[2], ZURINA MOHD HANAPI[3], AND AZIZOL ABDULLAH[4]
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia
Corresponding author: Fara Jamal (f4ra.jamal@gmail.com)

This work was support by UPM Journal Publication Fund, Universiti Putra Malaysia.

ABSTRACT In mobile cloud computing (MCC), data are stored and processed by more powerful computing platforms located in clouds to enhance the capability of mobile devices. Although MCC brings many benefits to users and organizations, security has become the main concern for MCC because it involves third party environment where data confidentiality is not a guarantee. To overcome this challenge, attribute-based encryption (ABE) technique, a promising cryptography method was introduced to support both data confidentiality and access control simultaneously. However, most of the ABE used single certification authority (CA) and disjoint attribute authority (AA) to verify user identity which can be a single point of failure. Therefore, a new approach was proposed in this research to deal with the single point of failure in the authority. Furthermore, the cache-based scheduling algorithm was upgraded to improve the response time of user jobs. The experimental results show that the approach performed better compared to its competitor in terms of reliability. Additionally, it significantly reduced the number of read operations compared to its counterpart.

INDEX TERMS Access control, attribute-based encryption, cache scheduling, mobile agent, mobile cloud computing.

I. INTRODUCTION

Previously, organizations stored their data on local servers. The members of the organization could access the data using mobile devices with specific software installed. To achieve security, the organizations had defined some access control policies. However, this trend of accessing organizational data has been changed where users can access organization data using their personal devices such as smart phones, tablets, and laptops from anywhere and at any time. With the introduction of cloud computing, organizations prefer to store their data on cloud to reduce their operating expenditure [1]. Cloud offers storage and computing resources with several benefits such as low cost, scalability, flexibility, software integration and many more. Furthermore, mobile cloud computing (MCC) was introduced to enable the data stored on cloud are accessible to mobile users anywhere and any time [2].

In MCC, data are stored and processed on cloud to enhance the capability of mobile device. Since it involves third party environment (cloud service provider) where data confidentiality is not a guarantee, security and privacy have become the main concern for MCC [3]. Furthermore, the data stored on public cloud can also be leaked if it is not stored in encrypted form [4]. To solve this problem, attribute-based encryption (ABE) was introduced in the literature. ABE is a promising cryptography method that supports both data confidentiality as well as access control [1], [5]–[7]. ABE schemes introduced in [8] and [9] are the most appropriate encryption systems where the ciphertext is related to the access structure, while the user's secret keys are related to the attribute sets. The decryption can only happen if the user is verified by two layers of authorities: the certification authority (CA) and attribute authority (AA). CA verifies user identity and grants a certificate to the verified user, while AA is responsible for storing and verifying dynamic attributes of the user and assigning partial keys based on the

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

provided credentials. However, the existing ABE techniques used single CA and multiple disjoint AA to verify user identity which can be a single point of failure.

Motivation : The existing access control can suffer from a single point of failure if the CA and one of the AAs fail [10]. The decision to allow or deny user access is made by the CA and AA based on the static attributes, but the integrity of the decision can be questioned if the authority fails. One of the main issues concerned in ABE are eliminating the single point of failure and reducing the computation overhead [11]. Compromised trusted authority in the ABE scheme will lead to certificate and key issuing problems [10], [12]. Consequently, the verification process cannot be done, and the user cannot obtain the required credentials to access the data. The average execution delay for the authority to verify the user will also increase especially during peak time where the cloud needs to entrain huge numbers of request. In this research, a new approach was proposed to deal with the single point of failure and reduce the average execution delay in authority.

Contributions: The major contributions of the proposed approach are as follows:

1) Solve the single point of failure related to authority by introducing a solution where the state of the authority is stored at a backup node and upon the failure of the authority, the backup node restarts the failed authority function from the backup state stored in its local storage.
2) Enable efficient access of data by improving the cache-based scheduling of user requests proposed by [4].

The paper is organized into the following sections: Section II highlights the existing methods in the literature related to ABE access control mechanism; The preliminary knowledge to understand the proposed scheme is presented in Section III. Section IV elaborates the proposed system. Section V describes the implementation of the proposed system. Section VI gives results and discussion. Lastly, Section VII and VIII presents the Security Analysis and Conclusion respectively.

II. RELATED WORK

Organization that implements MCC must have a strong authentication technique place. According to [13], the key security prevents data leakage which can control the employees' access to the organization's data. One standard authentication mechanism must be in place to protect the data from any unauthorized access [14].However, the authentication alone is not enough. It is also very crucial to encrypt data before placing it on a public cloud to avoid data leakage to third party service providers [15].

In order to tackle the aforementioned problems, [5], [7], [16]–[27] introduced authority agent for data encryption. The private and public keys are generated by the same entity. Furthermore, both users' private keys and ciphertexts are associated with a set of attributes or a policy over attributes. A user can decrypt a ciphertext if the private key matches the ciphertext. The solution is based on one authority to encrypt and decrypt the access control. The authority can be compromised when the user privacy breaches and it represents a single point of bottleneck on security.

The single authority technique was then improved by [28]–[30]. The authors used the joint zero secret sharing and distributed a key generator protocol. A secure threshold multiauthority fuzzy identity was contracted using this protocol without the use of central authority. In their technique, every authority can join or leave the system freely at any time and issue the secret key without communicating among themselves. The technique cannot prevent user collusion where two users can pool their decryption keys to generate decryption keys, and it depends on the standard complexity assumption [31].

References [1], [32]–[35] proposed multiple AAs but maintained a single CA in their solution. There is one CA that issues identity-related keys to users and *N* attribute authorities that manages attributes and issues attribute-related keys. User's keys from different AAs are linked together using global identifier (GID). Each AA will bind to one attribute for verification and all authority agents must be globally trustworthy, active, and uncorrupted throughout the system lifetime. The advantage of this proposed system is there is no single entity that controls encryption and decryption. Every attribute set is divided into multiple disjoint sets and each disjoint set is assigned to each authority and the authority will not get any information about user's attributes. However, it also increases computation and communication cost to maintain a fully trusted authority. Although the AAs are distributed, it still results in a single point of failure for CA. If the authority is being compromised, the entire system will fail, and key-escrow issue will be created. Based on the proposed solution by above researchers, the encryption process is based on the static attribute. It does not use mobile device dynamic attributes for encryption and decryption, which are not suitable for MCC. Thus, when the mobile device is stolen by an adversary, the privacy of the user may breach, and the data can be accessed in an unauthorized way [4].

Dynamic attribute encryption is the best way to overcome the above limitation and suitable for MCC environment. References [36], [37], and [38] introduced dynamic attribute that incorporated in mobile device using multiple authorities. In this scheme, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. However, this scheme does not preserve the user privacy. Authorities in the scheme need to communicate with each other to prevent collusion by combining their information to figure out any unauthorized information [11].

To improve the collusion issue, [15] introduced a scheme that preserves the user privacy and relies only on decisional bilinear Diffie–Hellman (DBDH) standard complexity assumption. This authority can work independently without any cooperation and central authority. The GID is used to tie all the user's secret keys together, while the corrupted authorities cannot pool the user's attributes by tracing it. During the startup phase, all authorities collaboratively setup the global environment. References [1] and [4] used the same technique but enhanced it with the implementation of client and server agents to sustain the communication in weak or disconnected network mode. The proposed approach deploys the pairs of client side agents (CSAs) that run on the user's mobile device while the server side agent (SSA) runs within the wired network to reduce data transmission over the wireless network. This technique can address the constraint of frequent disconnection and handle low bandwidth constraint, hence it sustains mobile computation even in a disconnection-prone mode. However, it also has a single point of failure in the authorities.

The above solutions show that there is limited literature that used dynamic attribute for encryption and decryption in their approaches. The existing approaches do not solve the single point of failure in CA and failed AAs. Thus, it is difficult to maintain the availability of data. In order to solve this problem, a new approach was proposed in this paper which eliminates the authority failure by supporting each authority (N) with a backup authority (BN). In this way, the failure of one or more nodes will not stop the system from providing the required services. Furthermore, there is also very limited literature that discusses on the use of cache to minimize execution delay for the authority to verify the attribute within the network. In this research, the caching algorithm introduced by [4] was enhanced to include scheduling for a better performance.

III. PRELIMINARIES

The preliminaries related to the proposed approach are described in this section. The proposed solution used a pair of agents to communicate between the CSA and SSA. To maintain user privacy, anonymous key issuing protocol and cache technique were used to optimize the data transmission.

A. CLIENT/SERVER SIDE AGENT

In order to address the issue related to mobile device connection with the cloud storage issue, the pair of CSA and SSA was used. CSA was installed in the user mobile device while SSA served as the local silent server. The CSA and SSA can perform mobile agent optimization to maintain the connection between mobile device and cloud storage during a weak connection. The process of creating CSA and SSA was adapted from [39]. The detail of the processes are as follows:

a) Init() function: Create new agent.
b) Clone() function: Copy agent.
c) Dispatch() function: Move agent to new host.
d) Retract() function: Bring back agent from Host B to A.
e) Deactive() function: Put the agent in sleep mode.
f) Activate() function: Resume the agent to its state.
g) Dispose() function: Delete the agent and its state.

B. BILINIAR PAIRING

Bilinear pairing was used to generate public keys from the AA.

Let F and FS be two distinct cyclic groups of prime order p and let $(m_1 \mid m_2)$ be the point generators in F. The bilinear map $\hat{e} : F_1 * F_2 \rightarrow F_S$ is said to be an admissible map if it fulfils the following conditions:

a) Bilinearity: There exists

$$\hat{e}\left(y^a, z^b\right) = \hat{e}\,(y, z)^{ab}, \quad \forall \text{a, b} \in Y_v \text{ and } \forall_y \in F_1,\ \forall_z \in F_2$$

b) Non-degeneracy: There exists

$$\hat{e}\,(y, z) \neq 1, \quad \forall y \in F_1,\ \forall_z \in F_2$$

c) Computable: There exists a proficient algorithm to evaluate

$$\hat{e}\,(y, z), \quad \forall y \in F_1,\ \forall_z \in F_2$$

C. LAGRANGE INTERPOLATION

The Lagrange interpolation was used to get a combined secret from the shared secrets. Suppose $m_u\,(y)$ is a polynomial of degree u, which satisfies $m_u\,(ya) = z_a$, where $1 \leq a \leq y$. The interpolating polynomial, $m_u\,(y)$ can be written as:

$$m_u\,(y) = \sum_{a=0}^{u} z_a \nabla_{u,a}(\text{y})$$

The polynomial $\nabla_{u,a}\,(y)$ is the Lagrange polynomial and $\{y_1, y_2, \ldots, y_u\}$ are the interpolation points. The polynomial $\nabla_{u,a}\,(y)$ can be defined as:

$$\nabla_{u,a}\,(y) = \prod_{b=0,b\neq a}^{u} \frac{y - y_b}{y_a - y_b}$$

Hence, the combined secret can be obtained as:

$$m\,(0) = \sum_{a=0}^{u} z_a \prod_{b=0,b\neq a}^{u} \frac{0 - y_b}{y_a - y_b}$$

D. ANONYMOUS KEY ISSUING PROTOCOL

The proposed solution used anonymous key-issuing protocol [39] to preserve user privacy. It is based on the anonymous credential system which supports non-linear coupling. The protocol consists of three entities as described:

a) CA which is a trusted authority in public key infrastructure based model. This authority verifies the identity of data owner and user from the CA list, then issues certificate to authentic user. This certificate will be encrypted using CA's secret key.
b) Key generation centre (KGC) generates encryption key to data owner or decryption key to user without knowing their identities to preserve anonymity. Multiple AA was used as KGC, where each authority has a pair of public and private keys.

c) To preserve user and data owner privacy, they need to present their identity to CA to get the credential. Then, the partial key is requested from each of the KGC. This partial key will be combined using the Lagrange interpolation, where a unique secret key will be issued to each data owner or user. Using this secret key, different pseudonyms are created to communicate with KGC. It is impossible for KGC to link the pseudonym with the owner and user profiles.

E. MULTI-AUTHORITY ATTRIBUTE BASED ENCRYPTION

The static and dynamic attributes were used to maintain data confidentiality in the proposed solution. The static attributes are divided into G number of disjoint set where each attribute is maintained by different AA. AA will publish public key and distribute encryption and decryption key after verifying user or owner static attribute. The steps involved are described as below:

GLOBAL SETUP

a) Global setup: The inputs provided to this algorithm are security parameters α and $\beta \in [0, 1]\, Poly\ (\alpha)$ which produce global parameters v_1 and v_2 as output which can be represented as:

$$GlobalSetup\left(1^{\alpha}1; \beta\right) \rightarrow v_1, v_2$$

b) Attribute key generator: In this step, each AA generates a pair of private and public keys for each static attribute. The i th AA randomly chooses the private key, v_i and the corresponding public key, y_i is calculated using the bilinear pairing mapping and the global parameters as:

$$y_i = \hat{e}\,(v_1, v_2)\, y_i$$

c) Encryption: The attribute set consists of static attributes, H and dynamic attributes, K. The data owner randomly chooses δ and $\theta \epsilon U_v$, and encrypts F as $Encrypt\,[F] = FY\theta_g$, where Y_g is the combined public key obtained from the partial public keys received from each individual AA by applying the Lagrange interpolation polynomial.

d) Decryption: The user downloads the required cipher-text and checks the associated access policies that must be satisfied to get the original message, F. For each Y_i, user requests the responsible AAs for the partial decryption credentials. Each AA verifies the associated static attributes and provides the corresponding decryption credential (private key), Y_i. The combined private key is obtained using the Lagrange interpolation polynomial denoted as Y_g, where k is the number of AAs.

F. ALGORITHM DATA CACHING

The cache function is to reduce the access time between the cloud and the device. The algorithm below shows the steps involve in data caching

```
Require: Query for data
Ensure: Cached data or retrieves data from the storage
server and cache it for future purpose
  if data is already available in the cache then
          return cached data.
  else
          search the requested data in the database.
          store data in the cache with cache validation.
          return data.
          end if
```

In the proposed approach, the RSA cryptographic algorithm and secure hash algorithm SHA-521 were used to maintain the integrity of the system. The integrity of certificate signed by CA and the dynamic attributes of the user's mobile device are essential in the proposed approach. The mobile agents perform caching to reduce the execution time. Thus, the total execution time of a mobile agent is diminished, and better performance is achieved.

IV. PROPOSED SOLUTION

The proposed technique is an extension of the work presented in [4] where the problem of the single point of failure has been tackled. Furthermore, the proposed technique also improves the response time for the incoming requests for data access at the cloud by using an improved cache-aware scheduling mechanism. A shared cache is proposed among several client requests running on the same host, hence improving the performance of the system.

The proposed architecture used the Cipher-text Policy ABE mechanism. The CA works as a certificate issuing agent and multiple AAs are responsible for maintaining the static attributes. The set of static attributes maintained by each AA is disjointed. These static attributes are used to verify the users before granting them the partial key for decryption.

In addition to that, the dynamic attributes are obtained from the mobile device of the user. An assumption was made similar to [4], where the logical sensors must be implemented in a way that can prevent malicious behavior of the user device.

In addition to the five entities (CA, multiple AAs, CSA, SSA, and CSS) proposed in [4], two additional entities were introduced at the authority side : (1) authority agent and cloud side: (2) request handler (RH) as illustrated in Figure 1. The authority agent is responsible to keep track of its neighbor node and send backup to the authority back node. The RH is responsible for handling the data access in an efficient way through cache-based scheduling.

A. CERTIFICATION AUTHORITY (CA)

CA is a certificate issuance authority which maintains a list of data owners and data users. In order to get a certificate from CA, the data owner or user must pass the verification process. This certificate is verified at the AA side when the user asks for encryption and decryption credentials from the AA.

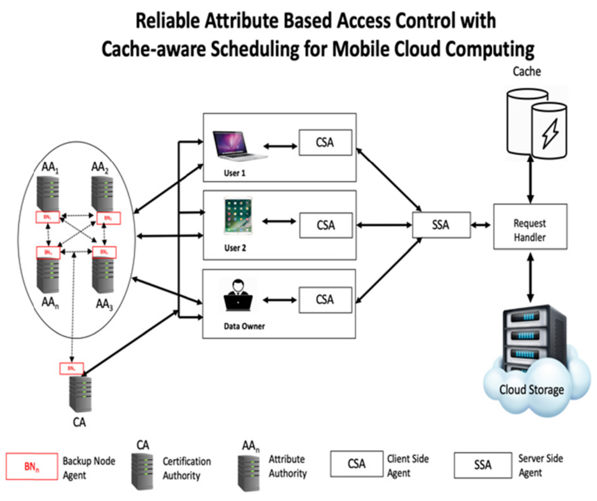


FIGURE 1. **Architecture of the proposed system.**

B. ATTRIBUTE AUTHORITY (AA)

AA is responsible for providing the encryption and decryption credentials to the owners/users certified by CA. Note that there are multiple AAs in the proposed architecture. The process at the AA side works as follows:

a) The data owner first selects a disjoint set of attributes from each AA. In response, each AA provides encryption credentials to the data owners. The data owner uses these credentials together with dynamic attributes to encrypt the data.
b) After encryption, the data is uploaded to the cloud to make it accessible to other users.
c) Now, to access the encrypted data, the user first provides the static attributes to each AA. After verifying the static attributes, AA responds with the decryption credentials. The user can decrypt the data if he/she has obtained the decryption credentials and the required dynamic attributes.
d) It is important to note that pseudonyms are used for data owners and users while communicating with the AAs to prevent malicious entities from identifying the real users.

C. CLIENT SITE AGENT (CSA)

User that requests to store or retrieve data from cloud storage go through the CSA. The CSA communicates with the SSA.

D. SERVER SITE AGENT (SSA)

The SSA handles user requests at the server side and acts as a surrogate of the user. The data sent and received by CSA and SSA are compressed to reduce latency.

E. CLIENT STORAGE SERVER (CSS)

The CSS provides storage and computation services to the data users and owners. In order to process the requests, virtual machines are created to provide the necessary storage and computational services.

F. AUTHORITY AGENT

Every authority node, N_i has a backup authority node, BN_i, which it selects from N. The agent running on node N_i keeps the backup of its state on the BN_i node by coordinating with the authority agent running on BN_i. During its operation, the agent at the N_i node will send updates of its state to the agent at its backup node BN_i, whenever the state of the agent at N_i changes. When the agent at N_i fails, the authority agent running at the BN_i node will recreate the agent from its backup state and will run it. In this way, failure of one or more nodes will not stop the system from providing the required services.

G. REQUEST HANDLER (RH)

The RH manages the shared cache memory and client requests for data access. Client requests are put in a priority queue where requests for data access are assigned with priority values based on the data in the data cache. Those requests' whole data that are already available in the cache are handled first to reduce the number of read operations. Request with required data that are not in the cache is processed later. It loads the data from the storage devices, provides it to the corresponding SSA, and puts the data in the cache memory for future use.

V. SYSTEM CONSTRUCTION

A. INTERNAL WORKING OF AUTHORITY AGENTS

In order to prevent the single point of failure for CA and AA, the mechanism as shown in Figure 2 was used. Consider n number of nodes denoted by $N = \{N_1, N_2, N_3, \ldots, N_n\}$. Every node Ni has a primary agent Ai that runs on N_i in a high priority thread. Furthermore, there can be a set of secondary agents running on N_i in low priority threads denoted by $S_i = \{s \parallel sE_n \wedge s! = A_i\}$. In addition, consider $BA_i = \{BA_1, BA_2, BA_3, \ldots., BA_k\}$ as the list of backup agents for A_i that every node N_i keeps for authority agents running on some other nodes such that A_i does not belong to BA_i. In the proposed strategy, every authority agent went through the five phases as illustrated by the activity diagram in Figure 2. Each phase is described in detail as follows:

a) **Setup Phase**: Authority agent running on node, N_i selects its backup node, BN_i from the set of nodes, N. In order to select the appropriate node from the set of nodes, the agent running on node N_i communicates with every other node and asks for the number of authority agents they are running and the number of backups they are holding. Then, using the following formula, the agent calculates the priority value for

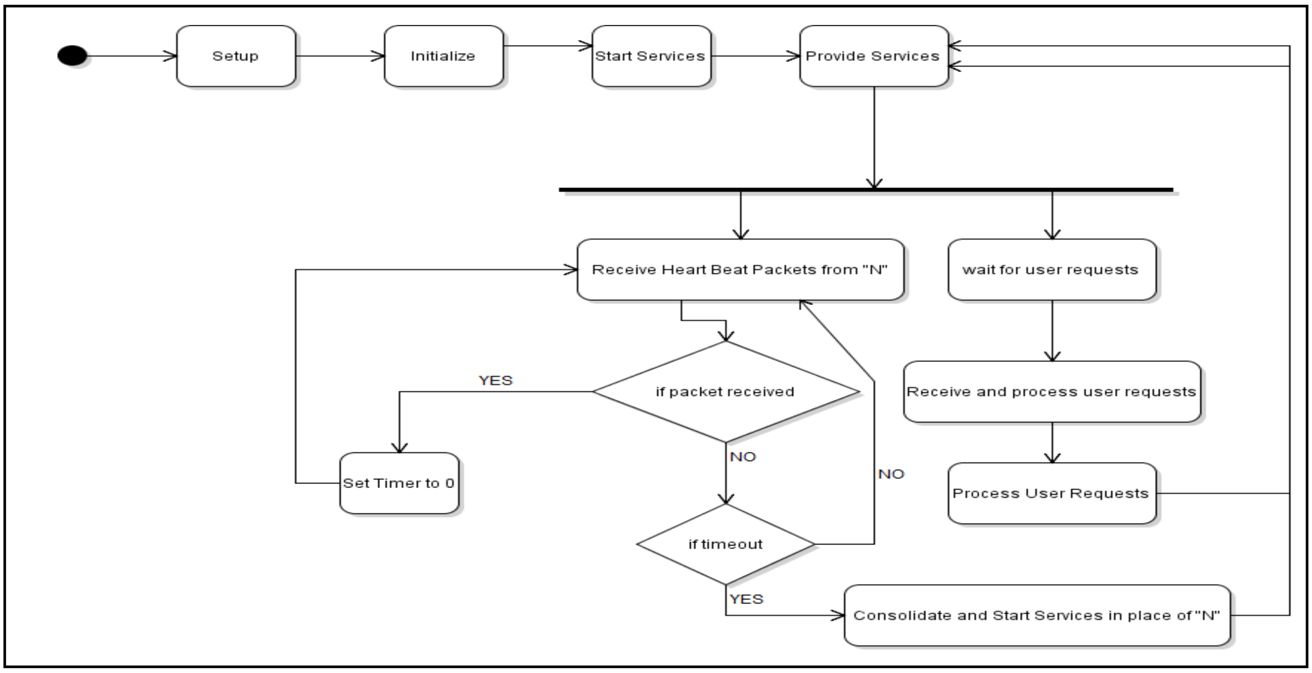


FIGURE 2. Internal working of the system.

every other agent.

$$P(N_i) = \frac{1}{Load\ on\ N_i}$$

$$Load\ on\ (N_i) = [A_i] + \frac{[B_i]}{[A_i] * WA}$$

$$Load\ on\ (N_i) = \frac{[A_i]^2 * WA + [A_i]\,[B_i]}{[A_i] * WA}$$

where *WA* denotes the weight of a single authority agents with respect to the backup kept by a node. After calculating the priority value for all the neighboring nodes, the N_i selects the nodes that have the highest priority value as a backup node, BN_i.

b) **Initialization phase**: All the authority agents running in their corresponding nodes initialize their services.

c) **Operation phase:** All the authority agents in their corresponding nodes provide services continuously to the end users. By services we mean providing credentials or partial keys based on the user provided static attributes (in case of AA agent) or providing certificates (in case of CA agent). Furthermore, the authority agent running on the current node requests the agent on its BN node to start the backup process in low priority thread so that it has minimum adverse effects on the performance of the main services of the agent running on the BN node.

d) **Update phase**: Authority agent running on node N_i sends update request about its state to the corresponding BN node. Upon receiving the request, the *BN* node starts a low priority thread to handle the update process. The main reason behind selecting the low priority thread is to reduce the effect of the update process on the services provided by the authority agent running on the current node. The thread automatically stops upon the completion of the update process. This process is necessary because the *BN* must have the updated state of the agent running on node *N* when it fails, to continue the services of the node *N* from where it stopped. The state refers to all the static attributes that the current node is responsible to handle for each user and the corresponding credentials that must be provided if the user is verified based on the provided static attributes.

e) **Consolidation Phase:** The authority agent running on *BN* node restores the failed agent from its backup and starts it. Upon starting, the backup node will be treated as a fresh node and goes through all the five phases as they went through at the beginning.

For example, there are two nodes each running one primary agent. The CA and AA are the two authority agents. Thus, CA will run on one node and AA run on the other node. During the setup phase, CA tries to find a backup node by communicating with each node and calculating the priority of each node using the equitation in the setup phase. Since there are only two nodes in this example, the CA will select the AA's node. CA communicates with AA and requests to keep its backup. AA accepts the request and puts the authority agent (CA) in its backup list. During its lifetime, if the state of the CA changes, it updates its state at the backup node. Furthermore, the CA regularly sends heartbeat message to the backup node to inform the backup node that it is working fine. When the backup node does not get any heartbeat packet from the corresponding node within the specified time period, it starts providing services in place of the failed CA node by restoring the failed agent from its backup. Now, the backup node is running two services: (1) CA and (2) AA. This will lead to performance degradation. However, the degradation is graceful compared to the existing strategies, where the system will fail to provide services if the CA fails.

B. USER VERIFICATION AND CERTIFICATION ISSUING

CA generates a pair of keys, where SK_r is the secret key and PK_r is the public key. CA will keep the secret key and publish the public key to be used by AA. The certificate is obtained by applying the RSA encryption algorithm over the hash value of CA's and requester's unique identities. The hash value is calculated using SHA-512 algorithm. The encryption and decryption of certificate can be represented as:

$$d\left[CA's\ ID \parallel Requester's\ ID\right] \rightarrow Certificate_t$$
$$Encrypt\ [Certificater]\, SK_r \rightarrow Certificate'_t$$

where t belongs to the set of data owner or user, such that $\left\{B_1, B_2 \ldots\ldots, B_t, \ldots., B_f\right\}$ or $C_1, C_2, \ldots.., t, \ldots., C_n$. Each B represents the data owner and C represents the user, f and n are the total number of data owners and users, respectively. The certified data owners or users request AAs for partial encryption or decryption credentials. The requested AA verifies the certificate by decrypting it with CA's public key.

C. GLOBAL SETUP

The input of the global setup algorithm is a security parameter (α and β). The output is the bilinear group and a global parameter, GP, where $(1^{\alpha}, \beta) \rightarrow GP$

for each attribute $i\,(1 \leq i \leq N)$ where N is the total number attribute monitor by the authority and i is the private keys known only to AA. AA generates a pair of private and public keys for each static attribute. The ith AA randomly chooses the private key, $\mathrm{Y_i}$ and the corresponding public key, Y_i is calculated using the bilinear pairing mapping and the global parameters which can be written as: $Y_i = \hat{e}\,(v_1, v_x)^{y_i}$.

D. ENCRYPTION

The encryption algorithm takes a set of attributes maintained by AA and a set of dynamic attributes from user device chosen by data owner. The algorithm will produce the ciphertext of the data as the output. Data owner will choose random private keys (δ and θ) and calculate corresponding public keys (T_0 and T_1). The hash value of the dynamic attribute will be in T_0.

The attribute set consists of static attributes, H and dynamic attributes, K. The steps involved are as follows:

The data owner randomly chooses δ and $\theta \in U_v$ and encrypts F as $Encrypt\ [F] = FY_g^{\theta}$, where $\mathrm{Y_g}$ is the combined public key obtained from the partial public keys received from each individual AA by applying the Lagrange interpolation polynomial. The hash value of the dynamic attributes is computed using SHA-512 algorithm as

$$\mathrm{K}' = \mathrm{d}\,[k_1 \parallel k_2 \parallel k_3 \ldots. \parallel k_c]$$

The data owner evaluates

$$T_0 = K'Y_g^{\delta+\theta}\ and\ T_1 = v_2^{\delta}$$

The data owner uploads

$$RS_F = [Encrypt\ [F]\,, T_0, T_1, Y_i \quad \forall 1 \leq i \leq g]$$

to the cloud storage server.

E. DECRYPTION

The user downloads the required ciphertext RS_F and checks the associated access policies that must be satisfied to get the original message, F.

The steps involved are:

For each Y_i, user requests the responsible AAs for the partial decryption credentials.

Each AA verifies the associated static attributes and provides the corresponding decryption credential (private key) Y_i. The combined private key is obtained using the Lagrange interpolation polynomial denoted as Y_k, where k is the number of AAs.

VI. ACCESS OF DATA THROUGH SSA USING CACHE BASED SCHEDULING

The Algorithm1 is used at the cloud server side where user requests arrive for data access. This algorithm uses the shared cache memory to enable efficient access to data. The algorithm takes the User Request Queue (UR) as input. It begins by sorting the UR at Line 1 based on the data that are already present in the cache memory. For instance, requests whose required data is already in the cache memory are processed first, while the user requests whose data is not in the cache are processed at the end. At Line 2, it takes each UR from the UR Queue, if the data required by the current user request is already in the cache (Line 3), it just fetches data directly from the cache (Line 4). Otherwise, it reads the data from the storage device and stores it in the cache for future use at Line 6.

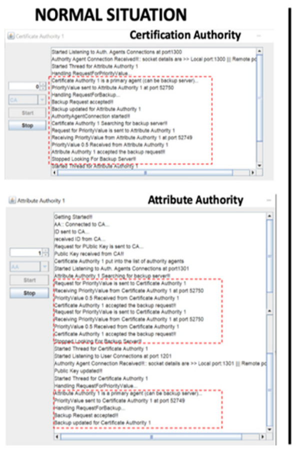
VII. SIMULATION RESULT AND DISCUSSION

In order to prove the effectiveness of the proposed architecture, the computational time of encryption and decryption in the proposed solution and the solution by [4] was simulated using Socket programming in Java. Java has built-in support for the cryptography. For ciphertext policy attribute-based encryption, the Java implementation presented by [40] was used. The solution by [4] was also simulated using the same context.

The simulation was executed on an Intel core i7 processor @2.50 GHz, 8 GB RAM running on a Windows 10 operating system and Eclipse IDE for Java developer 2019-03. Several

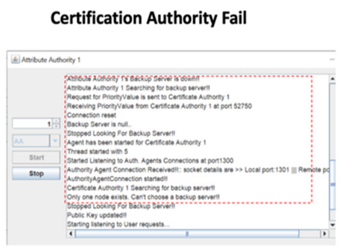
Algorithm 1 Improved Cache-Based Scheduling of User Requests

```
Algorithm Schedule_User_Requests
Input  UR = {UR_1, UR_2, UR_3, ..., UR_n}
BEGIN
1    SortQueueBasedOnCacheData ( UR );
2    FOR each User Request in UR
3      IF the data required by the User
                Request is in Cache
                Memory THEN
4        Read data from the cache;
5      ELSE
6        Fetch data from the storage
              and store it in the
              cache for future use;
7    END FOR
END
```



```
Started Listening to Auth. Agents Connections at port1300
Authority Agent Connection Received!!:: socket details are >> Local port:1300 ||| Remote po
Started Thread for Attribute Authority 1
Handling RequestForPriorityValue...
Certificate Authority 1 is a primary agent (can be backup server)...
PriorityValue sent to Attribute Authority 1 at port 52750
Handling RequestForBackup...
Backup Request accepted!!
Backup updated for Attribute Authority 1
AuthorityAgentConnection started!!
Certificate Authority 1 Searching for backup server!!
Request for PriorityValue is sent to Attribute Authority 1
Receiving PriorityValue from Attribute Authority 1 at port 52749
PriorityValue 0.5 Received from Attribute Authority 1
Attribute Authority 1 accepted the backup request!!
Stopped Looking For Backup Server!!
Started Thread for Attribute Authority 1
```

```
Getting Started!!
AA:: Connected to CA...
ID sent to CA...
received ID from CA...
Request for PUblic Key is sent to CA...
Public Key received from CA!!
Certificate Authority 1 put into the list of authority agents
Started Listening to Auth. Agents Connections at port1301
Attribute Authority 1 Searching for backup server!!
Request for PriorityValue is sent to Certificate Authority 1
Receiving PriorityValue from Certificate Authority 1 at port 52750
PriorityValue 0.5 Received from Certificate Authority 1
Certificate Authority 1 accepted the backup request!!
Request for PriorityValue is sent to Certificate Authority 1
Receiving PriorityValue from Certificate Authority 1 at port 52750
PriorityValue 0.5 Received from Certificate Authority 1
Certificate Authority 1 accepted the backup request!!
Stopped Looking For Backup Server!!
Started Thread for Certificate Authority 1
Started Listening to User Connections at port 1201
Authority Agent Connection Received!!:: socket details are >> Local port:1301 ||| Remote po
Public Key updated!!
Started Thread for Certificate Authority 1
Handling RequestForPriorityValue...
Attribute Authority 1 is a primary agent (can be backup server)...
PriorityValue sent to Certificate Authority 1 at port 52749
Handling RequestForBackup...
Backup Request accepted!!
Backup updated for Certificate Authority 1
```



```
Attribute Authority 1's Backup Server is down!!
Attribute Authority 1 Searching for backup server!!
Request for PriorityValue is sent to Certificate Authority 1
Receiving PriorityValue from Certificate Authority 1 at port 52750
Connection reset
Backup Server is null..
Stopped Looking For Backup Server!!
Agent has been started for Certificate Authority 1
Thread started with 5
Started Listening to Auth. Agents Connections at port1300
Authority Agent Connection Received!!:: socket details are >> Local port:1301 ||| Remote po
AuthorityAgentConnection started!!
Certificate Authority 1 Searching for backup server!!
Only one node exists. Can't choose a backup server!!
Stopped Looking For Backup Server!!
Public Key updated!!
Starting listening to User requests...
```

FIGURE 3. CA node fail.

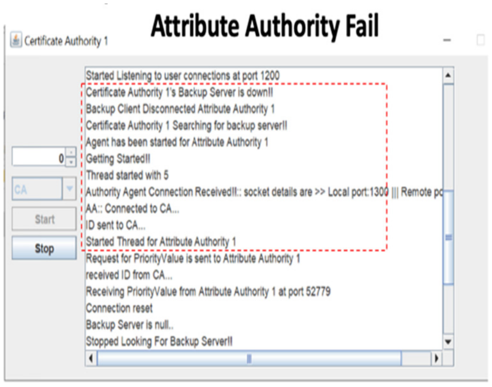
experiments were performed with the increasing of static and dynamic attributes varying from 1 to 10. The experimental total values of total time taken were recorded each time.

A. PERFORMANCE ANALYSIS AND EVALUATION

Several experiments were conducted to evaluate the performance of the proposed solution as described as follows:

1) EXPERIMENT 1: COMPUTATIONAL COMPLEXITY DURING AUTHORITY NODE FAIL

The effectiveness of the proposed strategy was analyzed and tested with different numbers of authority, in case when the CA and AA nodes fail. It started with one CA node and one AA node and failed the CA node. The strategy by [4] fails to provide services when CA down as the AAs cannot provide encryption and decryption credentials to the data owners and data users, respectively, without verifying their certificate. Since CA is not available to provide the certificate, AA will refuse to grant the encryption and decryption credentials to the data owners and data users. However, in the proposed solution, when the CA node fails (Figure 3), the CA backup



```
Started Listening to user connections at port 1200
Certificate Authority 1's Backup Server is down!!
Backup Client Disconnected Attribute Authority 1
Certificate Authority 1 Searching for backup server!!
Agent has been started for Attribute Authority 1
Getting Started!!
Thread started with 5
Authority Agent Connection Received!!:: socket details are >> Local port:1300 ||| Remote po
AA:: Connected to CA...
ID sent to CA...
Started Thread for Attribute Authority 1
Request for PriorityValue is sent to Attribute Authority 1
received ID from CA...
Receiving PriorityValue from Attribute Authority 1 at port 52779
Connection reset
Backup Server is null..
Stopped Looking For Backup Server!!
```

FIGURE 4. AA node fail.

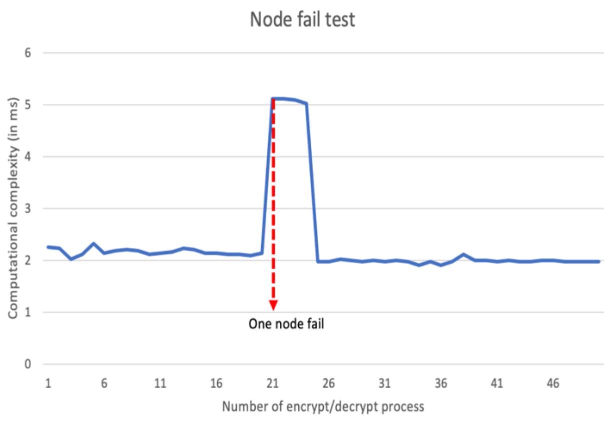


FIGURE 5. Node fail test

at the backup node is activated and started providing services. The CA is now running in the AA node. In this way, the proposed system is resistant to the single point of failure in case of single CA.

Then, the proposed solution was tested again using one CA node and one AA node but with a failed AA node. When the AA node fails, the AA backup at the backup node is activated and started providing services. Now the AA is running on the CA as shown in Figure 4. In this way, the proposed system is resistant to the single point of failure in case of single AA.

The experiment was conducted using different values of u chosen from the set of 1 to 50 incorporated u as the number of encryption and decryption process. In Figure 5, the performance of the system degraded as there were two services run by a single node, i. e., CA and AA. The degradation will only happen when the node started to initialize its process but then the certificate and the key generated by the authority is faster now because it is done by a single node.

Next, the experiment was conducted with different value of v ranging from 1 to 10, where v is the number of authority node running. The experiment started with $v = 10$ where v_1 run the CA function and v_2 to v_{10} run the AA functions. The node was disabled one by one to compare the result. Figure 6 shows in normal situation with 10 nodes running,

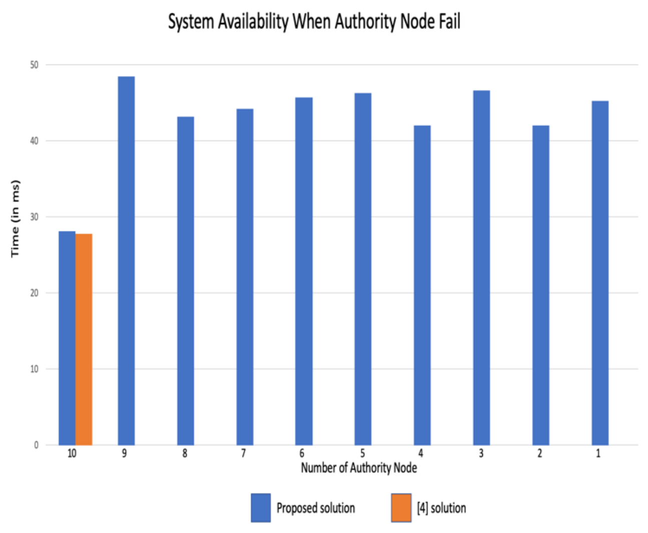


FIGURE 6. System availability when authority fail.

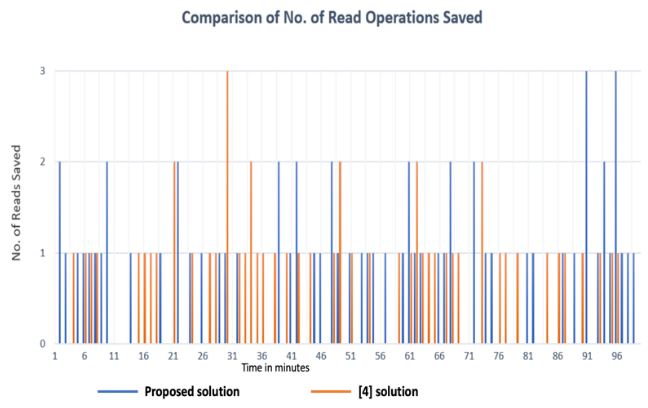


FIGURE 7. Number of read comparison.

the proposed solution and solution by [4] were able to produce encrypt and decrypt keys. The x-axis shows the time taken for the system to generate the encrypt and decrypt keys. When one node (v– 1) was disabled, the solution by [4] was unable to function. Meanwhile, the proposed solution functioned and maintained its service until one authority node was left. In this situation, the one node will perform CA function and nine AAs function.

2) EXPERIMENT 2: TOTAL NUMBER OF READ

The experiment was simulated using 10280 jobs to access data in the cloud and the total time taken was 481 s to complete. During that time, the total read saved for solution by [4] was 215 reads while the proposed solution managed to save 373 reads. Figure 7 shows the subset of the tested result within 100 min. The number of reads saved can be compared between the cache-based scheduling algorithm proposed in this work and the cache-based scheduling proposed by [4]. The x-axis shows the simulation time in minutes, while the y-axis shows the number of reads saved. It is observed that the number of read operations saved by the proposed cache-aware scheduling scheme was much higher than that of the cache-based scheduling scheme proposed by [4]. This is due to the fact that the approach by [4] is local to the SSA. Moreover, it does not consider other requests in queue for current data in the cache.

VIII. SECURITY ANALYSIS

In this section, the possible security threat to the proposed ABE architecture is explained. Each issue is addressed and the architecture reliable against those security threats is validated.

A. PROTECTION AGAINST MALWARE THREAT

To prevent from malware injected to the storage in the cloud, the proposed approach used CA to verify data owner. CA will maintain the list of authenticated data owner and user and will generate the certificated if the requester identity is valid. If the attacker tries to request for certificate, CA will verify that the attacker identity is not in the list of verified users, the request will be dropped by CA.

Example Situation: Let F_t = attacker that injects malware and tries to forge the certificate.

$$\boldsymbol{D[F'_tID] \rightarrow Certificate_t\ and\ Encrypt_{RSA}\ [Certificate_t]\,SK_t \rightarrow certificate'_t}$$

By using this certificate, F_t requests decryption credential form AAs. All the AA will decrypts the certificate using CA's public key $(\boldsymbol{PK_r})$ ***and decrypt*** $\boldsymbol{[Cerificate'_t]\,PK_c}$.

The certificate is not original from CA, so AA will discard the request.

Proof:

The proposed solution can protect against malware threat.

B. PROTECTION AGAINST IDENTITY THEFT THREAT

The pseudonyms were used instead of the user real identity in key-issuing protocol to communicate with AAs. This is to prevent the identity theft attack since the AAs cannot identify the identity of requested user or system owner.

Example Situation: To obtain the decrypt key:

User randomly chooses the private value $(\boldsymbol{v, \tau_1, \tau_2 \epsilon K_p})$,

AA randomly chooses the private value $(\boldsymbol{Y_h, v, Q_h \epsilon K_p})$.

User private value and AA private value will be combined:

$$\boldsymbol{(u,\tau_1,\tau_2)+(Y_{h,u},Q_h) \rightarrow X = (u+\beta_h)\,\tau_1\ mod\tau\& \\ Q = u + Y_h, u)\tau_2\ mod\tau}$$

The calculated value of x and y is passed to AA to obtain the decrypt key.

AA will calculate randomize decryption key using x and y information without knowing the user identity.

Proof: The proposed solution will hide user identity thus protect against identity theft.

C. PROTECTION AGAINST FAIL CA INCASE OF ATTACK

In a case of attacker trying to bring down the CA, the neighbor node will take over CA function to generate credential.

Example Situation: Backup node (*BN*) request respond from CA, N.

If there is no respond, BN will initialize its service, which BN will provide CA service until N is restored and joint the network.

Proof: The proposed solution will continue to perform encryption and decryption even in case of attack to CA by having the backup node to replace the CA function.

D. PROTECTION AGAINST COLLUSION ATTACK

In the ABE scheme, there will be two possible types of collusion attack which is (1) collusion between AA and accumulated the user attributes and (2) decryption keys are pooled together to access the data which cannot be accessed by individual users. The proposed solution can protect up to (N-2) AA.

Example Situation: To obtain the decrypt key: User randomly chooses the private value $(\boldsymbol{u}, \boldsymbol{\tau_1}, \boldsymbol{\tau_2} \in \boldsymbol{K_p})$.

The user identity, u incorporated within decryption key by inverse exponentiation operation, u will be added to the AA private value $(Y_{k,u}, Q_h \in K_p)$ to get the value of x and y. In the algorithm, it is infeasible to infer the value x and y.

Proof: The proposed solution can protect against collusion attack since it is impossible to modify the value of u with other user identity.

IX. CONCLUSION AND FUTURE WORK

In this paper, an agent-based ABE access control method was proposed for the mobile cloud environment. This type of access control is suitable to be implemented in organizations that allows its members to bring their own devices to access the organization data. The data are stored in the cloud in the encrypted form. User static attributes are used to provide encryption and decryption credentials to the end users. The data are encrypted or decrypted using the provided credentials along with the dynamic attributes of the user obtained from users' device. The existing strategies are suffering from the single point of failure problem where the failure of CA will stop the working of the whole system; as the users will not receive any certificate. Therefore, AA will not provide any encryption and decryption credentials. Hence, in this paper, the agent-based backup mechanism was proposed to overcome this issue. The approach also enhances the caching process by implementing the cache-based schedule. The experimental results show that in the proposed scheme, the system is still able to provide certificates to the users when the authority is at its original node fails. Furthermore, the cache-based scheduling algorithm significantly reduces the number of reads operations compared to its competitor. The security analysis of the proposed approach was also discussed.

Future work is needed to extend the solution to deal with securing the process of choosing the backup node.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their comments.

REFERENCES

[1] F. Li, Y. Rahulamathavan, M. Conti, and M. Rajarajan, "Robust access control framework for mobile cloud computing network," *Comput. Commun.*, vol. 68, pp. 61–72, Sep. 2015.
[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2011.
[3] Y. Jin, C. Tian, H. He, and F. Wang, "A secure and lightweight data access control scheme for mobile cloud computing," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Aug. 2015, pp. 172–179.
[4] N. Agrawal and S. Tapaswi, "A trustworthy agent-based encrypted access control method for mobile cloud computing environment," *Pervasive Mobile Comput.*, vol. 52, pp. 13–28, Jan. 2019.
[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2005, pp. 457–473.
[6] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. ICALP*. Berlin, Germany: Springer-Verlag, 2008, pp. 579–591.
[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.
[8] V. C. Hu, D. R. Kuhn, J. Voas, and D. F. Ferraiolo, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, Feb. 2015.
[9] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 126–138, Jan. 2013.
[10] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 858–880, 1st Quart., 2018.
[11] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Survey on revocation in ciphertext-policy attribute-based encryption," *Sensors*, vol. 19, no. 7, p. 1695, 2019.
[12] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Revocable, decentralized multi-authority access control system," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput. Companion (UCC Companion)*, Dec. 2018, pp. 220–225.
[13] N. Zahadat, P. Blessner, T. Blackburn, and B. A. Olson, "BYOD security engineering: A framework and its analysis," *Comput. Secur.*, vol. 55, pp. 81–99, Nov. 2015.
[14] S. Pao and M. Thorne. (2016). *BYOD Security: Expert Tips on Policy, Mitigating Risks, & Preventing a Breach Meet Our Panel of Data Security Experts*. Accessed: Sep. 18, 2018. [Online]. Available: https://digitalguardian.com/blog/byod-security-expert-tips-policy-mitigating-risks-preventing-breach
[15] S. Lee, D. Goel, E. L. Wong, and M. Dahlin, "Towards privacy-preserving bring-your-own-apps (BYOA)," Univ. Texas Austin, Austin, TX, USA, Tech. Rep., 2014.
[16] J. Lai, Y. Li, J. Weng, and R. H. Deng, "Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption," in *Proc. ASIA CCS*, Kyoto, Japan, Jun. 2014, pp. 239–248.
[17] I. Denisow, S. Zickau, F. Beierle, and A. Küpper, "Dynamic location information in attribute-based encryption schemes," in *Proc. 9th Int. Conf. Next Gener. Mobile Appl., Services Technol.*, 2015, pp. 240–247.
[18] S. G. Weber, "A hybrid attribute-based encryption technique supporting expressive policies and dynamic attributes," *Inf. Secur. J., Global Perspective*, vol. 21, no. 6, pp. 297–305, Apr. 2016.
[19] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu, and J. Qian, "A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram," *Proc. IEEE*, vol. 5, pp. 1137–1145, 2017.
[20] L. Ibraimi, M. Petkovic, S. I. Nikova, P. H. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," Centre Telematics Inf. Technol., Enschede, The Netherlands, Tech. Rep. TRCTIT-09-12, 2009.

[21] X. Liang, Z. Cao, D. Xing, and H. Lin, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proc. ASIACCS*, Sydney, NSW, Australia, Mar. 2009, pp. 343–352.
[22] K. Emura, A. Miyaji, A. Nomura, M. Soshi, and K. Omote, "Scheme with constant ciphertext length," *Int. J. Appl. Crypography*, vol. 2, no. 1, pp. 46–59, 2010.
[23] N. Attrapadung, B. Libert, and E. de Panafieu, "Encryption with constant-size ciphertexts," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2011, pp. 90–108.
[24] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 53–70.
[25] F. Li, D. Weerasinghe, D. Patel, and M. Rajarajan, "An user-centric attribute based access control model for ubuquitous environments," in *Mobile Computing, Applications, and Services*. Berlin, Germany: Springer, 2012, pp. 361–367.
[26] C.-J. Wang and J.-F. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext," in *Proc. 8th Int. Conf. Comput. Intell. Secur.*, 2012, pp. 447–451.
[27] L. Pang, J. Yang, and Z. Jiang, "A survey of research progress and development tendency of attribute-based encryption," *Sci. World J.*, vol. 2014, Jul. 2014, Art. no. 193426.
[28] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. CCS*, Chicago, IL, USA, Nov. 2009, pp. 121–130.
[29] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Progress in Cryptology—INDOCRYPT*. Berlin, Germany: Springer, 2008, pp. 426–436.
[30] M. U. Aftab, M. A. Habib, N. Mehmood, M. Aslam, and M. Irfan, "Attributed role based access control model," in *Proc. Conf. Inf. Assurance Cyber Secur. (CIACS)*, 2015, pp. 83–89.
[31] Y. Rahulamathavan, S. Veluru, J. Han, F. Li, M. Rajarajan, and R. Lu, "User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2939–2946, Sep. 2016.
[32] V. Božović, D. Socek, R. Steinwandt, and V. I. Villányi, "Multi-authority attribute-based encryption with honest-but-curious central authority," *Int. J. Comput. Math.*, vol. 89, no. 3, pp. 268–283, 2012.
[33] J. Li, Q. Huang, X. Chen, S. S. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, 2011, pp. 386–390.
[34] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2011, pp. 568–588.
[35] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1484–1496, May 2016.
[36] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 11, pp. 2150–2162, Nov. 2012.
[37] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Public-Key Cryptography—PKC*. Berlin, Germany: Springer, 2013, pp. 125–126.
[38] Z. Liu, Z. Cao, Q. Huang, T. H. Yuen, and D. S. Wong, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," in *Computer Security—ESORICS*. Berlin, Germany: Springer, 2011, pp. 278–297.
[39] N. Agrawal and S. Tapaswi, "Access control framework using dynamic attributes encryption for mobile cloud environment," in *Progress in Advanced Computing and Intelligent Engineering* (Advances in Intelligent Systems and Computing). Singapore: Springer, 2018, pp. 611–621.
[40] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2112–2120.

FARA JAMAL received the B.Sc. degree in information technology (networking) and the M.Sc. degree in information technology majoring in security from Universiti Utara Malaysia, in 2001 and 2005, respectively. She is currently pursuing the Ph.D. degree in security in computing with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. In 2004, she joined one of the Ministry in Malaysian Public Sector in Putrajaya, where she is a System Analyst and incharge of the Data Centre, Network and Security, for more than 13 years. Her research interests include the area of access control in mobile cloud computing and BYOD security.

MOHD TAUFIK ABDULLAH received the Ph.D. degree from Universiti Putra Malaysia, Malaysia. He is currently an Associate Professor with the Department of Computer Science and a member of the Information Security Research Group, Universiti Putra Malaysia, Malaysia, where he is also the Programme Coordinator for Master of Information Security with the Department of Computer Science, Faculty of Computer Science and Information Technology. His research interest includes security in computing and digital forensics.

ZURINA MOHD HANAPI received the B.Sc. degree in computer and electronic system engineering from Strathclyde University, in 1999, the M.Sc. degree in computer and communication systems engineering from Universiti Putra Malaysia (UPM), in 2004, and the Ph.D. degree in electrical electronic and system engineering from the Universiti Kebangsaan Malaysia, in 2011. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, UPM, where she is also a Lecturer. She is also a Leader of some research projects. She has authored many conference and journal articles. Her research interests include routing, wireless sensor networks, wireless communication, distributed computing, network security, cryptography, and intelligent systems. She received the Excellence Teaching Award, in 2005, 2006, and 2012, the Silver Medal, in 2004, and the Bronze Medal, in 2012.

AZIZOL ABDULLAH received the M.Sc. degree in engineering (telematics) from The University of Sheffield, U.K., in 1996, and the Ph.D. degree in distributed system from Universiti Putra Malaysia, Malaysia, in 2010. He is currently an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. He has been a Fellow Researcher with ITU-UUM Asia Pacific Centre of Excellence for Rural ICT Development (ITU-UUM). His main research areas include cloud and grid computing, network security, wireless and mobile computing, and computer networks.