International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6359
***Article Received**: 10 January 2023 **Revised**: 14 February 2023 **Accepted**: 25 February 2023*

BLA2C2: Design of a Novel Blockchain-based Light-Weight Authentication & Access Control Layer for Cloud Deployments

Surbhi Khare[1], Dr. Abhishek Badholia[2]
[1]Research Scholar, Department of CSE,
MATS School of Engineering & IT
Raipur, India
surbhikhare20@gmail.com
[2]Associate Prof., Department of CSE,
MATS School of Engineering & IT
Raipur, India
drabhishekb@matsuniversity.ac.in

Abstract: Cloud deployments are consistently under attack, from both internal and external adversaries. These attacks include, but are not limited to brute force, masquerading, improper access, session hijacking, cross site scripting (XSS), etc. To mitigate these attacks, a wide variety of authentication & access control models are proposed by researchers, and each of them vary in terms of their internal implementation characteristics. It was observed that these models are either highly complex, or lack in terms of security under multiple attacks, which limits their applicability for real-time deployments. Moreover, some of these models are not flexible and cannot be deployed under dynamic cloud scenarios (like constant reconfigurations of Virtual Machines, dynamic authentication use-cases, etc.). To overcome these issues, this text proposes design of a novel blockchain-based Light-weight authentication & access control layer that can be used for dynamic cloud deployments. The proposed model initially applies a header-level light-weight sanitization layer that removes Cross Site Scripting, SQL Injection, and other data-level attacks. This is followed by a light-weight authentication layer, that assists in improving login-level security for external attacks. The authentication layer uses IP matching with reverse geolocation mapping in order to estimate outlier login attempts. This layer is cascaded with an efficient blockchain-based access control model, which assists in mitigating session hijacking, masquerading, sybil and other control-level attacks. The blockchain model is developed via integration of Grey Wolf Optimization (GWO) to reduce unnecessary complexities, and provides faster response when compared with existing blockchain-based security deployments. Efficiency of the model was estimated in terms of accuracy of detection for different attack types, delay needed for detection of these attacks, and computational complexity during attack mitigation operations. This performance was compared with existing models, and it was observed that the proposed model showcases 8.3% higher accuracy, with 10.5% lower delay, and 5.9% lower complexity w.r.t. standard blockchain-based & other security models. Due to these enhancements, the proposed model was capable of deployment for a wide variety of large-scale scenarios.

Keywords: Cloud, Access, Authentication, Sybil, Internal, External, Delay, Accuracy, Complexity, Attacks.

1. Introduction

Access control and selective ownership modelling is a multidomain process that includes the design of control rules, ownership groups, key-exchange techniques, and secure storage models. This process also includes the design of control rules. The responsibility of regulating and permitting improved entity-level access to user nodes that need cloud services falls on control rules. The nodes that are able to traverse these rules without error are grouped together into ownership groups, and the access level of each individual node is either granted or refused for each individual entity. The linking of user accounts with cloud services utilizing user-control, group-control, and role-control layers results in the creation of a virtual private access (VPA) layer. These levels are controlled by ownership and access control rules. A sample example of the implementation of access control measures for cloud infrastructure that is supported by Amazon Web Services is shown in Figure 1. (AWS). [1] This architecture establishes separate user roles for the AWS compute services and the AWS Internet of Things (IoT) services. Within the context of this paradigm, users have access to a portion of the Main AWS stack by way of a particular internal rule layer that is part of the AWS IoT stack. Dual rule mapping is the method that cloud service providers use in order to provide access to a certain user group for a portion of the services that they offer.

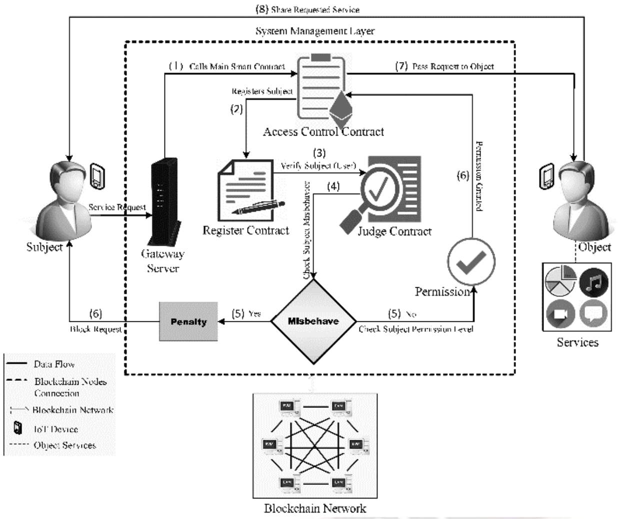

Figure 1. A typical cloud-level model for access control & authentication operations [1]

Based on this model, it can be observed that a Standard access control and ownership architecture requires the development of efficient rule engines, as well as group control layers, registration layers, and secure storage layers. For this purpose, a wide variety of system models are recommended [2, 3, 4], each of which is unique in terms of the level of security it provides, the quality of service it provides, the efficiency with which it enforces access control, its capacity to scale, and so on. The next section will give a brief assessment of a few of the most recent approaches for access control and selective ownership enforcement. This will be followed by a discussion of the complexities of these systems, as well as their merits and downsides. Based on this review, it is observed that these models are either highly complex, or lack in terms of security under multiple attacks, which limits their applicability for real-time deployments. Moreover, some of these models are not flexible and cannot be deployed under dynamic cloud scenarios (like constant reconfigurations of Virtual Machines, dynamic authentication use-cases, etc.). To overcome these issues, next section of this text proposes design of a novel blockchain-based Light-weight authentication & access control layer that can be used for dynamic cloud deployments. The model was evaluated on multiple use cases, and was compared with existing state-of-the-art techniques in terms of accuracy of attack detection, delay needed for attack analysis, and computational complexity levels. Finally, this text concludes with some interesting contextual observations about the proposed models and also recommends methods to further optimize its performance for different deployments.

2. Literature Review

A wide variety of access control and authentication models are proposed by researchers, and each of them vary in terms of their real-time performance under different cloud scenarios. For instance, work in [5, 6] propose use of Scalable Attribute-Based Access Control, and Symmetric Searchable Encryption (SSE) with Attribute-Based Encryption (ABE), which assist in improving security of cloud deployments. But these models are not scalable for large-scale clouds. To overcome these issues, work in [7] proposes use of Fine-Grained Cloud Access Control, which deploys low complexity layers for highly cloud deployments. Similar models are discussed in [8, 9, 10] which propose use of Activity control (ACON), Reputation Centres (RC), and hierarchical key access mechanisms, which aim at lowering complexity of deployments via optimizing security & QoS levels for different attack types. Extensions to these models are discussed in [11, 12, 13], which recommend use of Stability-Based Controllers, ciphertext-policy attribute-based encryption (CP-ABE), and Tenant-Led Ciphertext Information Flow Control (TLC IFC), but these models have higher complexity due to integrated attack removal and encryption operations. Models that propose use of Secure and Efficient Multiauthority Access Control (SEMAC) [14], Context-Aware Policy Enforcement (CAPE) [15], Bidirectional Access Control [16], ordered binary decision diagram (OBDD) [17], Multi-keyword Ranked Search scheme with Fine access control (MRSF) [18], Revokable ABE [19], and multi-authority attribute-based encryption (MA-ABE) [20] are discussed by researchers for different use cases. These models aim at integrating general purpose layers for different attack types, which makes them highly efficient under real-time scenarios.

Similar models are discussed in [21, 22, 23, 24, 25], which propose use of Traceable Attribute-Based Encryption Scheme with Dynamic Access Control (TABE DAC), Hybrid CP ABE, Tactile Networks, extended file hierarchy CP-ABE scheme (EFH-CP-ABE), and Server-Aided Fine-Grained Access Control (SAF GAC), that aims at incorporating multilevel access control mechanisms in order to improve attack detection capabilities for different cloud types. These models are extended in [26, 27, 28, 29, 30], wherein researchers have proposed use of improved ABE, Non-singular Terminal Sliding Mode Control, Native Components, Trust Management, and Anonymous Distributed Fine-Grained Access Control Protocol with Verifiable Outsourced Decryption (VOD ADAC), that aims at integrating incremental learning mechanisms to support different cloud infrastructures under real-time deployments for multiple attack scenarios. Similar to these, the work in [31, 32, 33, 34, 35, 36] proposes use of Decentralized Attribute Based Access Control, Secure Attribute-Based Access Control With Identical Sub-Policies, Time and Attribute Factors Combined Access Control for Time-Sensitive Data samples, Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems, ordered binary decision diagram (OBDD), and privacy-preserving, revocable ciphertext policy attribute-based encryption (PR-CP-ABE) that can be applied under different real-time applications with higher degree of control and privacy options for high attack use cases. These models must be validated for larger clouds and can be extended via Smart-Contract-Based Access Control [37], Server-Aided Bilateral Access Control [38], Sanitizable Access Control System [39], Enhanced Access Control [40], Intelligent Role-Based Access Control Model [41], Secure Deduplication with User-Defined Access Control [42], and Lightweight and Expressive Fine-Grained Access Control [43] that utilize lightweight layers for filtering requests under multilevel deployed cloud types. But these models are either highly complex, or lack in terms of security under multiple attacks, which limits their applicability for real-time deployments. Moreover, some of these models are not flexible and cannot be deployed under dynamic cloud scenarios (like constant reconfigurations of Virtual Machines, dynamic authentication use-cases, etc.). To overcome these issues, next section of this text proposes design of a novel blockchain-based Light-weight authentication & access control layer that can be used for dynamic cloud deployments. The model was validated under different use cases, & attack scenarios, and its performance was compared w.r.t. standard cloud deployment models under real-time use cases.

3. Design of the proposed novel Blockchain-based Light-weight Authentication & Access Control layer for Cloud deployments

Based on the review of existing authentication & access control models used for cloud deployments, it can be observed that these models are either highly complex, or lack in terms of security under multiple attacks, which limits their applicability for real-time deployments. Moreover, some of these models are not flexible and cannot be deployed under dynamic cloud scenarios (like where VMs are constantly reconfigured, dynamic authentication use-cases, etc.). To overcome these issues, this section proposes design of a novel blockchain-based Light-weight authentication & access control layer that can be used for dynamic cloud deployments. Flow of the model is depicted in figure 2, where it can be observed that the proposed model initially applies a header-level light-weight sanitization layer that removes Cross Site Scripting, SQL Injection, and other data-level attacks.

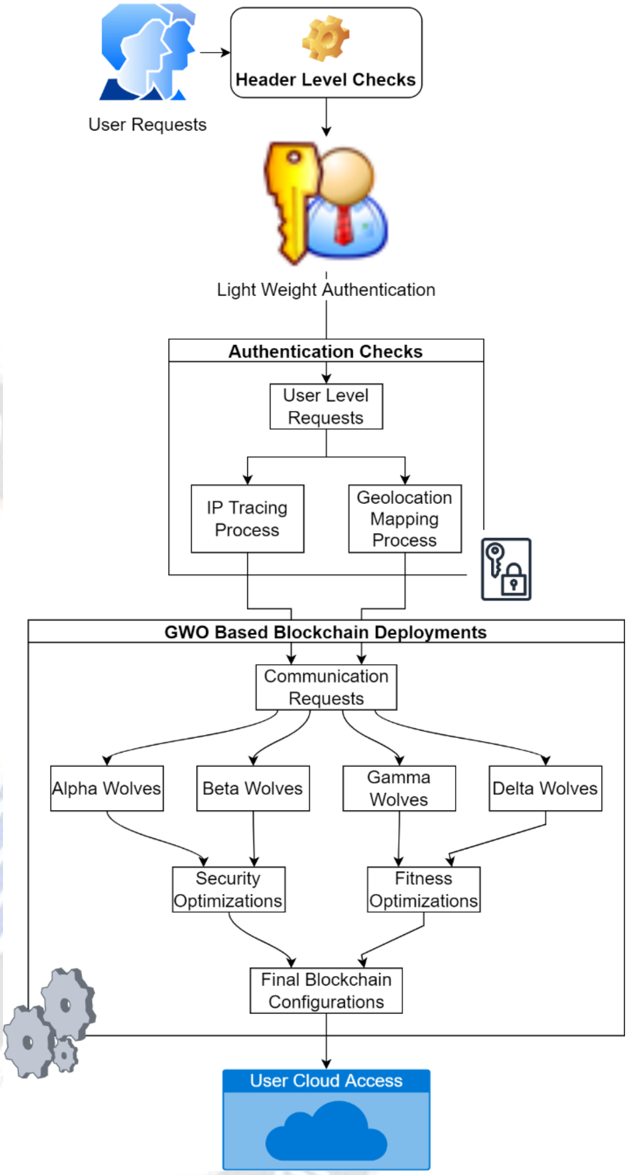

Figure 2. Flow of the proposed authentication model with customized blockchains

This is followed by a light-weight authentication layer, that assists in improving login-level security for external attacks. The authentication layer uses IP matching with reverse geolocation mapping in order to estimate outlier login attempts. This layer is cascaded with an efficient blockchain-based access control model, which assists in mitigating session hijacking, masquerading, sybil and other control-level attacks. The blockchain model is developed via integration of Grey Wolf Optimization (GWO) to reduce unnecessary complexities, and provides faster response when compared with existing blockchain-based security deployments. The model design is segregated into 3 sub modules, and each of them are discussed in separate sub-sections of this text.

3.1. Design of the header-level checks layer for initial request filtering operations

Each request sent by cloud users is initially scrutinized through a series of light-weighted pattern checks. The computational complexity of these checks is very low; thus, they require an infinitesimal delay for checking requests. Each of the requests is passed through the following checks,

Check & remove presence of $< or >$ symbols, which might interact with cloud scripts.

Remove all $'$ quote & $--$ commend characters for filtering out SQL & external script attacks

Replace the keywords, wscript, seekSegmentTime, mocha, vbs, view-source, xmlns:xdp, applescript, jar, livescript, form, form:action,Javascript, FSCommand, behavior, style, xlink:href, vbscript, or jscript, with blank characters to avoid any script level attacks

Remove all characters with encoding other than UTF16, which ensures that any invisible characters are not processed by the cloud deployments.

IPs of the requests that are flagged with presence of the given characters are stored on the server (in the form of sessions) and reported to the administrator with every authentication request. This assists in improving authentication attack detection performance for external adversaries, and also assists in temporal blocking of unwanted users. Design of this authentication layer is discussed in the next section of this text.

3.2. Design of the authentication checks layer with user-level filtering operations

All filtered requests are processed via an authentication checker layer that uses user id and password combination (or any other password-based authentication can also be used), for logging in the users. After each login request is sent by cloud users, the following processed is used for authentication checks,

A unique session ID is assigned to each user, which is stored on the server for future checks

For each login, user's access levels are checked, and if user is accessing a restricted resource, then their IP is reported to admins

All requests that pass these rules are allowed into the cloud, else they are redirected to login panel

For continuously requesting users, their IPs are blocked from the cloud

After login, a temporal validity metric is estimated for each of the IPs via equation 1,

$$T_{valid_{t_1,t_2}} = \left(\frac{\sum_{T=t_1}^{t_2} R_{invalid_T}}{\sum_{T=t_1}^{t_2} R_{valid_T}}\right) \ \dots (1)$$

Where, $T_{valid_{t_1,t_2}}$ is the validity metric between the time intervals t_1 & t_2, $R_{invalid_T}$ are count of invalid requests during the same interval sets, while R_{valid_T} are total valid requests during the time intervals. For IP checks, a validity threshold (V_{th}) is estimated via equation 2,

$$V_{th} = \beta * \frac{\sum_{t=t_1}^{t_2} T_{valid_i}}{t_2 - t_1} \dots (2)$$

Where, β is a dynamic validity factor, which is updated as per equation 3,

$$\beta = \frac{\left(\begin{matrix}\sum_{i=1}^{N_c} N_{i_{valid_{t_1,t_2}}} \\ -\frac{\sum_{i=1}^{N_c} N_{i_{invalid_{t_1,t_2}}}}{N_c}\end{matrix}\right)}{N_c} \dots (3)$$

Where, $N_{i_{valid_{t_1,t_2}}} and\ N_c$ are the valid requests between time interval t_1, t_2, and total requests sent by user during the given time intervals. The value of β is re-evaluated for each login request, and then an IP-level rank is calculated via equation 4,

$$R_{E,IP} = \frac{\sum_{t_1=1}^{N} \sum_{t_2=t_1+1}^{N} \frac{B_{R_{t_1,t_2}}}{T_{R_{t_1,t_2}}}}{t_1 * (t_2 - 1)} \dots (4)$$

Where, $B_{R_{t_1,t_2}}$ are number of requests that were blocked during the time interval t_1, t_2, while $T_{R_{t_1,t_2}}$ are number of processed requests during the same time interval between t_1, t_2. After each login, a request threshold is estimated via equation 5,

$$R_{th} = \gamma * \frac{\sum_{i=1}^{N_{Ent}} \sum_{j=1}^{N_{IP}} R_{i,j}}{N_{ent} * N_{IP}} \dots (5)$$

Where, $N_{ent}, \& \ N_{IP}$ are counts of entities accessed by the user & total IPs via which the user has previously logged in, while γ is a request constant which is estimated via equation 6,

$$\gamma = \gamma_{old} + \frac{Blocked_{IPs} - Attack_{IPs}}{N_{IP}} \dots (6)$$

Where, $Blocked_{IPs}, \& \ Attack_{IPs}$ are total blocked requests & attack requests from given IP addresses. All the IPs with rank more than R_{th} are blocked from accessing the cloud, while others are passed for cloud access. To perform this task, a User-level and IP-level graph is generated, which maps each User ID with Client IP as per figure 3 as follows,

***Article Received**: 10 January 2023 **Revised**: 14 February 2023 **Accepted**: 25 February 2023*

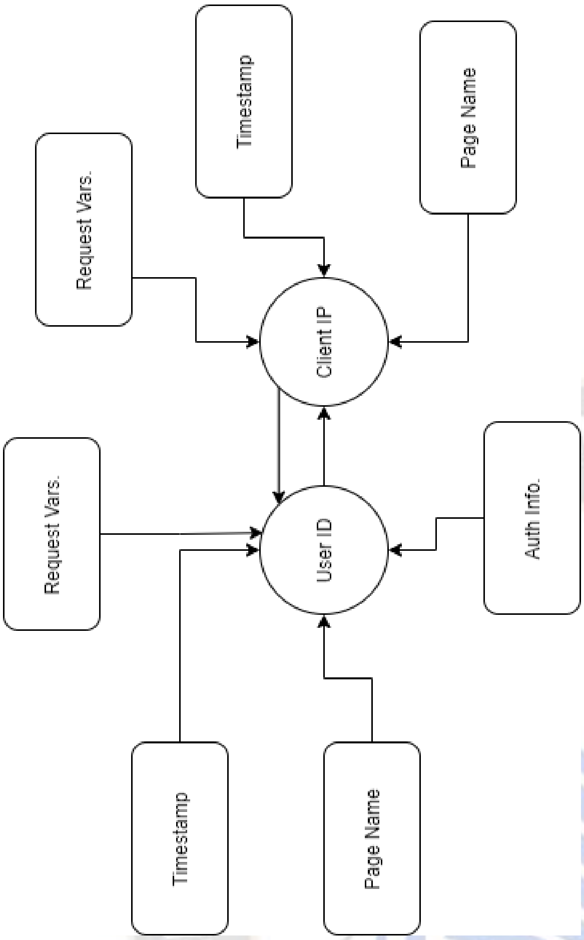

Figure 3. The IP and User ID mapping graph for authentication operations

Based on this graph, administrators are able to visualize different users and map them according to the IP address and access patterns. These access patterns are processed via a GWO based access control layer, which stores these patterns on customized blockchains. Design of this model is depicted in the next section of this text.

3.3. Design of the GWO based blockchain layer for efficient access control operations

Each access request is passed through a simple rule engine, where access is granted as per equation 7,

$$A_p^{IP}(U) = 1, when\ U^{IP} \in P \ldots (7)$$

Where, A_p^{IP} represents access grant status for page P and each of the user IPs. Each of these grant statuses are stored on a blockchain, which is optimized via GWO based optimizations. To perform this optimization, a constant representing number of blocks in each blockchain is estimated via equation 8,

$$N(Blocks)^{GWO} = \sum_{i=1}^{N_{sc}} \frac{L_i}{N_{sc}} \ldots (8)$$

Where, $N_{sc}, \& L_i$ are current number of sidechains (parts of blockchain generated by the GWO process), and length of each of these chains. Using this value, perform the following GWO based optimization operations,

- Initialize optimization parameters for the GWO as follows,
  - Total optimization iterations decided by the cloud designer (N_i^{GWO})
  - Total Wolves that are setup by the cloud designer (N_w^{GWO})
  - Cognitive learning rate for each of the Wolves (L_r^{GWO})
- Setup all Wolves as ‘Delta’ Wolves, and perform the following operations for N_i^{GWO} iterations,
  - Scan all Wolves, and modify each ‘Delta’ Wolf as per the following process,
    - From the list of sidechains, select a chain stochastically, and initiate dummy communication requests for this chain
    - Stochastically segregate requests into attack and normal requests
    - Evaluate the delay needed to process the attack & normal requests via equations 9 and 10 as follows,

$$D(A) = \frac{\sum_{j=1}^{A_{requests}} t_{end_j} - t_{start_j}}{A_{requests}} \ldots (9)$$

$$D(N) = \frac{\sum_{j=1}^{N_{requests}} t_{end_j} - t_{start_j}}{N_{requests}} \ldots (10)$$

Where, $A_{requests} \,\&\, N_{requests}$ are total attack and normal dummy requests, while $t_{start} \,\&\, t_{end}$ are time stamps at which these requests were sent to the cloud for processing operations. Now estimate attack and normal throughput levels via equations 11 and 12 as follows,

$$T(A) = \sum_{i=1}^{A_{requests}} \frac{Rx(P)_i}{D(A) * A_{requests}} \ldots (11)$$

$$T(N) = \sum_{i=1}^{N_{requests}} \frac{Rx(P)_i}{D(N) * N_{requests}} \ldots (12)$$

Where, $Tx(P) \,\&\, Rx(P)$ are total bytes transmitted and received during each of the requests. A combination of these metrics is used to estimate security level of the user as per equation 13,

$$SL_i = \frac{\frac{D(Normal)}{D(Malicious)} + \frac{T(Malicious)}{T(Normal)}}{2} \ldots (13)$$

***Article Received**: 10 January 2023 **Revised**: 14 February 2023 **Accepted**: 25 February 2023*

- These metrics are estimated for each of the sidechains, and then a Wolf fitness is estimated via equation 14,

$$f_w = \frac{\left[\sum_{i=1}^{N_s} SL_i - \sum_{j=1}^{N_s} \frac{SL_j}{N_{stoch}}\right]}{N_s} * \left[\frac{D(Normal) - D(Malicious)}{D(M)} + \frac{T(Malicious) - T(Normal)}{T(Normal)}\right] \dots (14)$$

  - Perform this task for each sidechain, and then estimate Wolf fitness threshold via equation 15 as follows,

$$f_{th} = \frac{1}{N_w} * \sum_{i=1}^{N_w} f_{w_i} * L_r \dots (15)$$

- Once an iteration is completed, then Wolf status is updated as per the following process,
  - Wolves with $f < \frac{f_{th}}{2}$ are marked as 'Alpha' Wolves
  - Wolves with $f < L_r * f_{th}$ are marked as 'Beta' Wolves
  - Wolves with $f < f_{th}$ are marked as 'Delta' Wolves
  - Wolves with $f \geq f_{th}$ are marked as 'Gamma' Wolves
- For each iteration, select a stochastic 'Alpha' Wolf, and update learning rate via equation 16,

$$New(L_r^{GWO}) = Min\left(\bigcup_{i=1}^{N_w^{GWO}} f_{w_i}^{GWO}\right) * \frac{Old(L_r^{GWO})}{\sum_{i=1}^{N_w^{GWO}} f_{w_i}^{GWO}} \dots (16)$$

After completion of N_i^{GWO} iterations, the 'Alpha' Wolves are used for either splitting or merging operations. To perform these operations, a set of dynamic rules is evaluated as per table 1, where C_B represents fitness of current blockchain, while A^{GWO} represents fitness of the 'Alpha' Wolf, which is used for these decisions.

Level of QoS (Eqn. 14)	Level of Security (Eqn. 13)	Decision taken for QoS & Security Levels
$C_B(QoS) > A^{GWO}(QoS)$	$C_B(SL) = A^{GWO}(SL)$	Merge the current chain with largest chain
$C_B(QoS) = A^{GWO}(QoS)$	$C_B(SL) = A^{GWO}(SL)$	Don't change blockchain configuration
$C_B(QoS) < A^{GWO}(QoS)$	$C_B(SL) = A^{GWO}(SL)$	Split smallest chain into 2 equal parts
$C_B(QoS) > A^{GWO}(QoS)$	$C_B(SL) > A^{GWO}(SL)$	Merge the current chain with largest chain
$C_B(QoS) = A^{GWO}(QoS)$	$C_B(SL) > A^{GWO}(SL)$	Merge the current chain with largest chain
$C_B(QoS) < A^{GWO}(QoS)$	$C_B(SL) > A^{GWO}(SL)$	Split smallest chain into 2 equal parts
$C_B(QoS) > A^{GWO}(QoS)$	$C_B(SL) < A^{GWO}(SL)$	Split smallest chain into 2 equal parts
$C_B(QoS) = A^{GWO}(QoS)$	$C_B(SL) < A^{GWO}(SL)$	Split smallest chain into 2 equal parts
$C_B(QoS) < A^{GWO}(QoS)$	$C_B(SL) < A^{GWO}(SL)$	Split smallest chain into 2 equal parts

Table 1. Set of dynamic rules for creation of blockchains

Once the blockchain is split into 2 equal parts, then the chain with higher SL is selected for adding new blocks. Based on this process, new blockchains are created and access control requests are stored on it for traceability purposes. Due to which, the model is highly secure, and can be used for real-time cloud deployments. Perform this model is evaluated in the next section of this text.

4. Results analysis and comparison

The model uses a combination of low complexity header-level checks along with IP based authentication & attack analysis. This is combined with a GWO based blockchain management process, which assists in improving its QoS & security performance under different attacks. The model was developed to function as a generic signup mechanism, and it requires a login, password, a verification photo, and more information in order to complete the registration process. It was planned out to function as a management system for electronic healthcare records (EHR). The user has the option of logging in as a "patient," "doctor," or "administrator." The administration has the ability to establish rules for access control, validate the blockchain, manage patients and doctors, inspect logs, and perform analysis on them. Patients are able to submit their reports, provide viewing access to physicians, search for physicians who are requesting ownership access, and grant or withdraw ownership access while checking in using their credentials and a numeric captcha (and transfer requests to public blockchain). Doctors are able to log in using their credentials and a numeric captcha, check for patient records that

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6359
***Article Received**: 10 January 2023 **Revised**: 14 February 2023 **Accepted**: 25 February 2023*

have read access, ask patients to take ownership of reports at the report level (and add the data to a private blockchain), and alter reports for which the patients have given them access. All of these actions can be performed after successfully logging in from different account types.

The model was evaluated on a total of 15 million login & access control requests. These requests were segregated as 15% authentications, 45% doctor access requests, 30% patient access requests, and 10% admin requests. For each of the requests, 30% were attack requests, while remaining 70% were normal requests. These requests were processed, and the following parameters were estimated for CP ABE [12], CAPE [15], and VOD ADAC [30], which assisted in comparison,

- Authorization attack accuracy (A3)
- Access control attack accuracy (ACAA)
- User input attack accuracy (UAA)
- Ownership attack accuracy (OAA)
- Authorization attack delay (A2D)
- Access control attack delay (ACAD)
- User input attack delay (UAD)
- Ownership attack delay (OAD)

Based on this evaluation strategy, the Authorization attack accuracy (A3) for different Number of Test Requests (NTR) was evaluated in figure 4, where the accuracy was averaged in order to estimate its true value under real-time use cases,

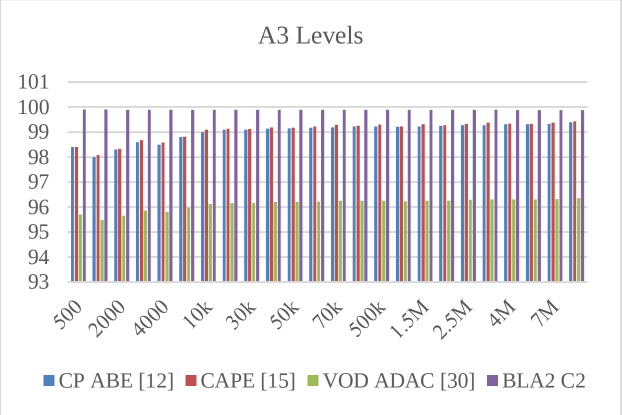

Figure 4. Authorization attack accuracy (A3) for different model scenarios

As per the observations in figure 4, it can be evaluated that the proposed model is capable of showcasing 1.5% better A3 levels than CP ABE [12], 2.5% better than CAPE [15], and 4.5% better than VOD ADAC [30], which makes it highly useful for identification of authentication attacks under real-time use cases. This accuracy is improved because of the light-weight attack detection layer, and use of highly efficient authentication models that can classify attacks with high accuracy levels. Similar observations were made for Authorization attack delay (A2D), and can be observed from Figure 5 as follows,

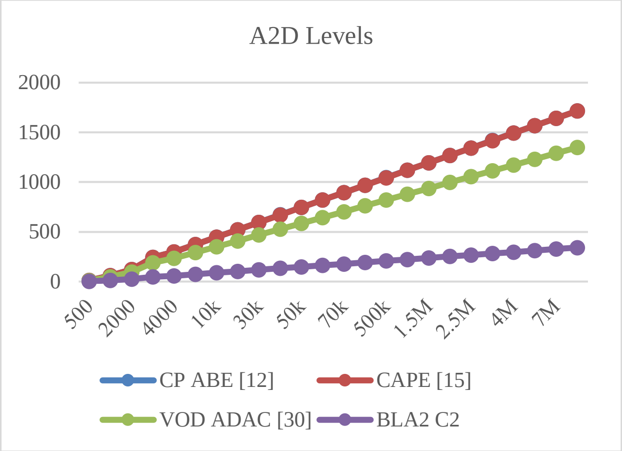

Figure 5. Authorization attack delay (A2D) for different model scenarios

As per the observations in figure 5, it can be evaluated that the proposed model is capable of showcasing 15.4% faster A2D performance than CP ABE [12], 19.4% faster A2D performance than CAPE [15], and 16.5% faster A2D performance than VOD ADAC [30], which makes it highly efficient for high-speed identification of authentication attacks under real-time use cases. This speed is improved because of the light-weight attack detection layer, and use of delay during modelling the GWO process. Similar observations were made for Access control attack accuracy (ACAA), and can be observed from Figure 6 as follows,

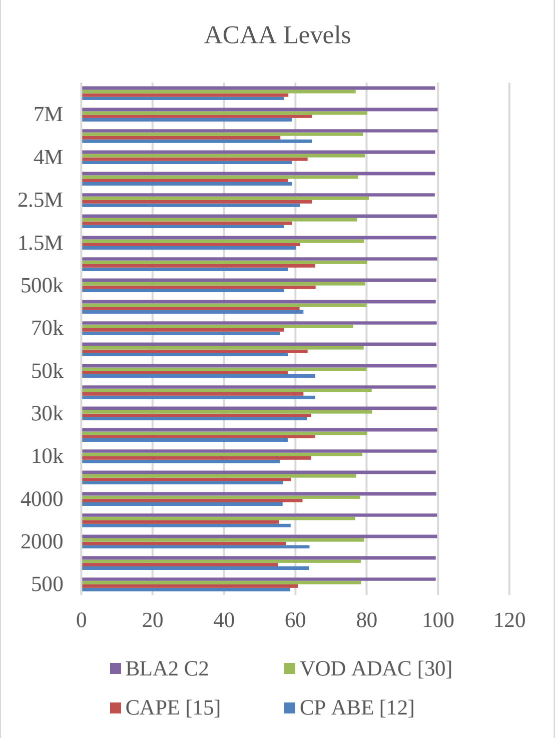

Figure 6. Access control attack accuracy (ACAA) for different model scenarios

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6359
***Article Received**: 10 January 2023 **Revised**: 14 February 2023 **Accepted**: 25 February 2023*

As per the observations in figure 6, it can be evaluated that the proposed model is capable of showcasing 2.5% higher ACAA performance than CP ABE [12], 1.8% higher ACAA performance than CAPE [15], and 3.5% higher ACAA performance than VOD ADAC [30], which makes it highly efficient for high-accuracy identification of access control attacks under real-time use cases. This accuracy is improved because of the light-weight attack detection layer, and use of low complexity access control operations. Similar observations were made for Access control attack delay (A2D), and can be observed from Figure 7 as follows,

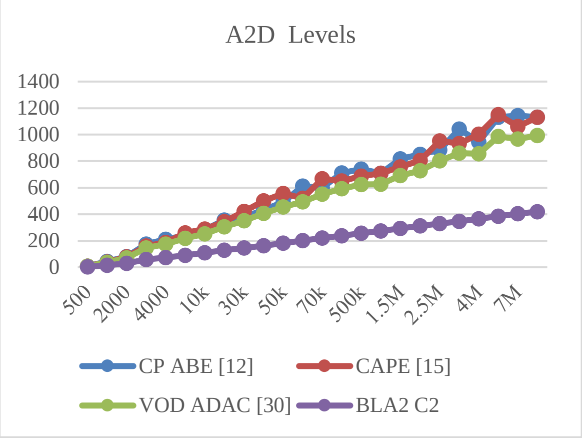

Figure 7. Access control attack delay (A2D) for different model scenarios

Based on this study, it can be shown that the recommended model performs better for real-time deployments since its access control check latency is 45% lower than CP ABE [12] & CAPE [15] and 20% lower than VOD ADAC [30]. This is due to the fact that request monitoring, together with the relatively small weight header level constraints, has improved the speed of its detection. Similar findings were found for the User input attack accuracy (UAA), which includes cookie hijacking, SQL injection, access control, and cross-site scripting (XSS). Figure 8 illustrates these vulnerabilities as follows,

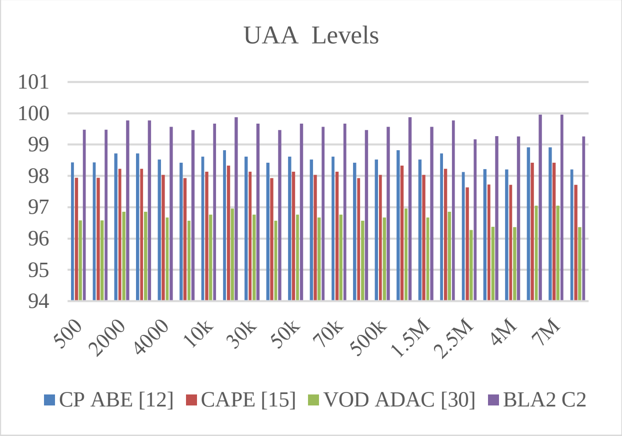

Figure 8. User input attack accuracy (UAA) for different model scenarios

In light of the findings of this research, it is possible to demonstrate that the model being proposed has a user attack detection accuracy that is 1.5%, 2.3%, and 2.9% more accurate than CP ABE [12], CAPE [15], and VOD ADAC [30]. This is as a result of the use of pattern analysis as well as header level rules. This enables the model to be deployed in real-time cloud environments by requiring it to accept requests only after they have successfully completed the pattern checks that are specified. Similar results were discovered regarding the User Input Attack Delay (UAD), and the following is seen in Figure 9,

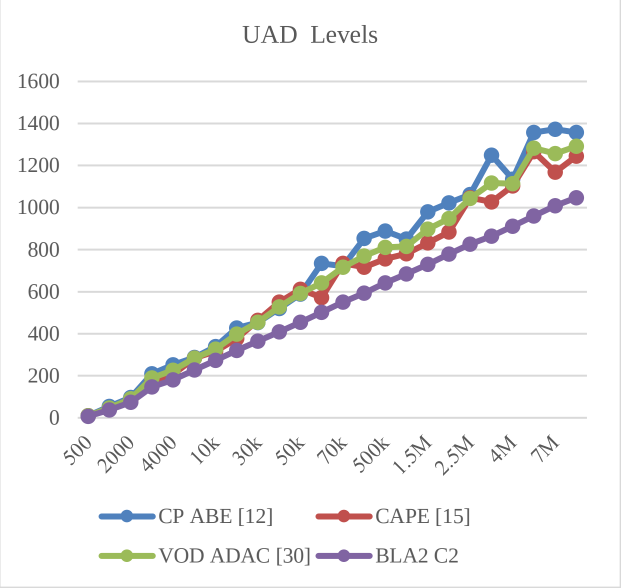

Figure 9. User input attack delay (UAD) for different model scenarios

On the basis of this study, it can be shown that the proposed model identifies user assaults with 25%, 18%, and 20% less latency than CP ABE [12], CAPE [15], and VOD ADAC [30], respectively; this demonstrates the model's increased performance for real-time deployments. This is due to the implementation of a lightweight pattern analysis engine at the header level, which has led to the current state of affairs. Injecting data modification packets and performing an analysis of invalid ownership requests were both found to provide results that were comparable to those observed for ownership attack accuracy (OAA) in figure 10 as follows,

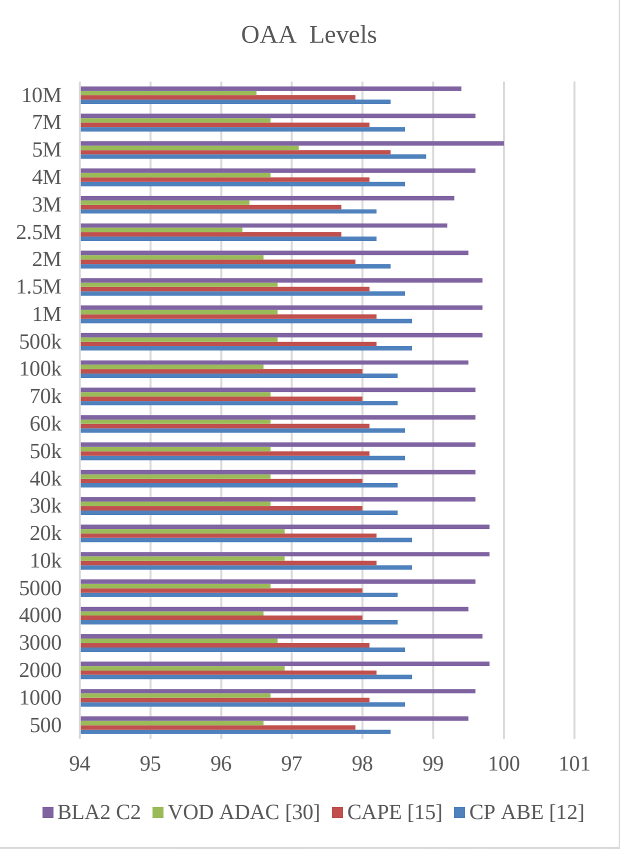

Figure 10. Ownership attack accuracy (OAA) for different model scenarios

In light of the investigation's results, it is evident that the recommended model has 1% better ownership assault detection accuracy than CP ABE [12], 1.5% better ownership assault detection accuracy than CAPE [15], and 3.1% better ownership assault detection accuracy than VOD ADAC [30]. This is the outcome of applying header level rules, pattern analysis, and a blockchain model created by a consortium. This enables the model to be deployed in real-time cloud environments by allowing it to only accept ownership queries after they have been handled by the given blockchain model. Similar findings were found in relation to the Ownership attack delay (OAD), and the findings are shown as follows in figure 11,

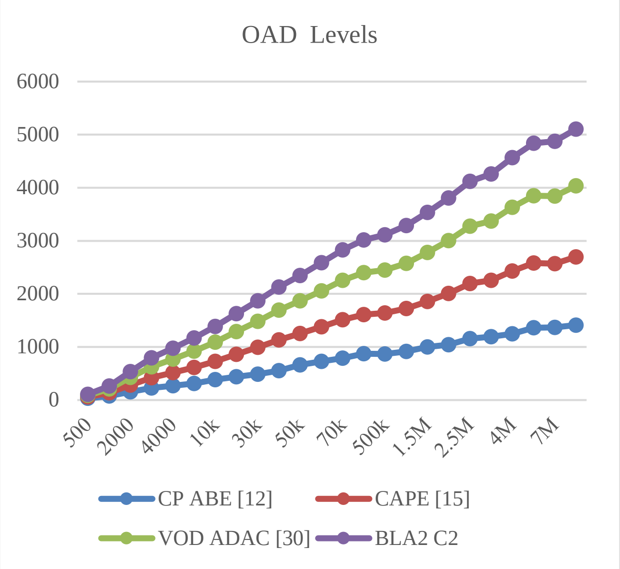

Figure 11. Ownership attack delay (OAD) for different model scenarios

On the basis of this study, it has been shown that the proposed model can identify ownership assaults 20% quicker than CP ABE [12], 10% faster than CAPE [15], and 15% faster than VOD ADAC [30]. This demonstrates the model's superior performance for real-time deployments. This is due to the fact that the GWO based sidechains, which only stores requests that have been completed, enables quicker detection of ownership attack detection operations under real-time use cases.

5. Conclusion and future scope

The proposed model is able to combine IP-based authentication and threat analysis with simple checks at the header level. Combining this with a blockchain-based GWO management mechanism improves its QoS and security performance in the face of a variety of threats. Comparing the model to other models revealed that the proposed model can achieve A3 levels that are 1.5%, 2.5%, and 4.5% better than CP ABE [12], CAPE [15], and VOD ADAC [30], respectively. This makes it very useful for detecting authentication attacks in real-world application cases. The deployment of extremely effective authentication models and the lightweight attack detection layer, which can effectively classify attacks, increases precision. In a similar vein, it was established that the suggested model has 15.4% faster A2D performance than CP ABE [12], 19.4% faster A2D performance than CAPE [15], and 16.5% faster A2D performance than VOD ADAC [30], making it very useful for high-speed detection of authentication assaults in real-time use scenarios. Performance is enhanced by the introduction of a lightweight attack detection layer and a delay during the modelling of the GWO process.

The proposed model provides access control performance that is 2.5% greater than CP ABE [12], 1.8% higher than CAPE

[15], and 3.5% higher than VOD ADAC [30]. This makes it very useful for detecting access control attacks with great precision in real-time application situations. This accuracy has risen due to the inclusion of low-complexity access control actions and a lightweight attack detection layer. It can also be shown that the proposed model beats CP ABE [12] & CAPE [15] and VOD ADAC [30] in terms of real-time deployments, with access control check latency that is 45% lower than CP ABE [12] & CAPE [15] and 20% lower than VOD ADAC [30]. This is due to the fact that request monitoring has accelerated its discovery, and the relatively lightweight header level constraints have also assisted.

In the presence of user level assaults, it was determined that the proposed model's user attack detection accuracy is 1.5%, 2.3%, and 2.9% more accurate than CP ABE [12], CAPE [15], and VOD ADAC [30]. This is because header-level rules and pattern analysis are applied. By requiring the model to accept requests only after they have successfully completed the supplied pattern checks, the model may be deployed in real-time cloud environments. Noting that the proposed model differentiates user assaults from CP ABE [12], CAPE [15], and VOD ADAC [30] with 25%, 18%, and 20% decreased latency, respectively, demonstrates the model's enhanced performance for real-time deployments. This is due to the implementation of a lightweight pattern analysis engine at the header level, which led to the current scenario.

When ownership of records was analyzed, it was discovered that the proposed model had a detection accuracy for ownership assaults that was 1% better than CP ABE [12], 1.5% better than CAPE [15], and 3.1% better than VOD ADAC [30]. This is the outcome of using pattern analysis, header level rules, and a consortium-developed blockchain model. This permits the model to be deployed in real-time cloud environments by allowing it to accept ownership queries only after they have been processed by the chosen blockchain model. In contrast, it has been shown that the proposed model can identify ownership assaults 20 percent quicker than CP ABE [12], 10 percent faster than CAPE [15], and 15 percent faster than VOD ADAC [30]. This demonstrates the performance of the model in real-time deployments. This is due to the fact that GWO-based sidechains may detect ownership attack operations in real-time usage circumstances more rapidly since they only store requests that have been properly completed.

In future, researchers can validate the model under different cloud deployments, and also can be extended via use of bioinspired security models like Genetic Algorithm, Bacterial Foraging Optimization, etc. The model’s performance can also be extended via integration of deep learning methods like Convolutional Neural Networks (CNNs), Autoencoders (AEs), etc. under real-time scenarios.

References

[1] S. Bhatt, T. K. Pham, M. Gupta, J. Benson, J. Park and R. Sandhu, "Attribute-Based Access Control for AWS Internet of Things and Secure Industries of the Future," in IEEE Access, vol. 9, pp. 107200-107223, 2021, doi: 10.1109/ACCESS.2021.3101218.

[2] S. Fugkeaw, "A Fine-Grained and Lightweight Data Access Control Model for Mobile Cloud Computing," in IEEE Access, vol. 9, pp. 836-848, 2021, doi: 10.1109/ACCESS.2020.3046869.

[3] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao and K. Yu, "AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud," in IEEE Access, vol. 8, pp. 70604-70615, 2020, doi: 10.1109/ACCESS.2020.2985762.

[4] S. A. Chaudhry, K. Yahya, F. Al-Turjman and M. -H. Yang, "A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems," in IEEE Access, vol. 8, pp. 139244-139254, 2020, doi: 10.1109/ACCESS.2020.3012121.

[5] R. Ahuja and S. K. Mohanty, "A Scalable Attribute-Based Access Control Scheme with Flexible Delegation cum Sharing of Access Privileges for Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 32-44, 1 Jan.-March 2020, doi: 10.1109/TCC.2017.2751471.

[6] A. Bakas, H. -V. Dang, A. Michalas and A. Zalitko, "The Cloud we Share: Access Control on Symmetrically Encrypted Data in Untrusted Clouds," in IEEE Access, vol. 8, pp. 210462-210477, 2020, doi: 10.1109/ACCESS.2020.3038838.

[7] G. Ra, D. Kim, D. Seo and I. Lee, "A Federated Framework for Fine-Grained Cloud Access Control for Intelligent Big Data Analytic by Service Providers," in IEEE Access, vol. 9, pp. 47084-47095, 2021, doi: 10.1109/ACCESS.2021.3067958.

[8] J. Park, R. Sandhu, M. Gupta and S. Bhatt, "Activity Control Design Principles: Next Generation Access Control for Smart and Collaborative Systems," in IEEE Access, vol. 9, pp. 151004-151022, 2021, doi: 10.1109/ACCESS.2021.3126201.

[9] L. Gao, Z. Yan and L. T. Yang, "Game Theoretical Analysis on Acceptance of a Cloud Data Access Control System Based on Reputation," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1003-1017, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2016.2632110.

[10] B. Celiktas, I. Celikbilek and E. Ozdemir, "A Higher-Level Security Scheme for Key Access on Cloud Computing," in IEEE Access, vol. 9, pp. 107347-107359, 2021, doi: 10.1109/ACCESS.2021.3101048.

[11] S. Guan and S. Niu, "Stability-Based Controller Design of Cloud Control System With Uncertainties," in IEEE Access, vol. 9, pp. 29056-29070, 2021, doi: 10.1109/ACCESS.2021.3059766.

[12] S. Guan and S. Niu, "Stability-Based Controller Design of Cloud Control System With Uncertainties," in IEEE Access, vol. 9, pp. 29056-29070, 2021, doi: 10.1109/ACCESS.2021.3059766.

[13] Z. Zhang, Z. Yang, X. Du, W. Li, X. Chen and L. Sun, "Tenant-Led Ciphertext Information Flow Control for Cloud Virtual Machines," in IEEE Access, vol. 9, pp. 15156-15169, 2021, doi: 10.1109/ACCESS.2021.3051061.

[14] S. Xiong, Q. Ni, L. Wang and Q. Wang, "SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 2914-2927, April 2020, doi: 10.1109/JIOT.2020.2963899.

[15] Y. Verginadis et al., "Context-Aware Policy Enforcement for PaaS-Enabled Access Control," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 276-291, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2927341.

[16] J. Cui, B. Li, H. Zhong, G. Min, Y. Xu and L. Liu, "A Practical and Efficient Bidirectional Access Control Scheme for Cloud-Edge Data Sharing," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 2, pp. 476-488, 1 Feb. 2022, doi: 10.1109/TPDS.2021.3094126.

[17] K. Edemacu, B. Jang and J. W. Kim, "Efficient and Expressive Access Control With Revocation for Privacy of PHR Based on OBDD Access Structure," in IEEE Access, vol. 8, pp. 18546-18557, 2020, doi: 10.1109/ACCESS.2020.2968078.

[18] J. Li, J. Ma, Y. Miao, R. Yang, X. Liu and K. -K. R. Choo, "Practical Multi-Keyword Ranked Search With Access Control Over Encrypted Cloud Data," in IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 2005-2019, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3024226.

[19] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei and K. -K. R. Choo, "CryptCloud$^+$+: Secure and Expressive Data Access Control for Cloud Storage," in IEEE Transactions on Services Computing, vol. 14, no. 1, pp. 111-124, 1 Jan.-Feb. 2021, doi: 10.1109/TSC.2018.2791538.

[20] K. Huang, "Secure Efficient Revocable Large Universe Multi-Authority Attribute-Based Encryption for Cloud-Aided IoT," in IEEE Access, vol. 9, pp. 53576-53588, 2021, doi: 10.1109/ACCESS.2021.3070907.

[21] L. Guo, X. Yang and W. -C. Yau, "TABE-DAC: Efficient Traceable Attribute-Based Encryption Scheme With Dynamic Access Control Based on Blockchain," in IEEE Access, vol. 9, pp. 8479-8490, 2021, doi: 10.1109/ACCESS.2021.3049549.

[22] S. Qi, Y. Lu, W. Wei and X. Chen, "Efficient Data Access Control With Fine-Grained Data Protection in Cloud-Assisted IIoT," in IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2886-2899, 15 Feb.15, 2021, doi: 10.1109/JIOT.2020.3020979.

[23] N. Gholipoor, S. Parsaeefard, M. R. Javan, N. Mokari, H. Saeedi and H. Pishro-Nik, "Resource Management and Admission Control for Tactile Internet in Next Generation of Radio Access Network," in IEEE Access, vol. 8, pp. 136261-136277, 2020, doi: 10.1109/ACCESS.2020.3011466.

[24] J. LI, N. CHEN and Y. ZHANG, "Extended File Hierarchy Access Control Scheme with Attribute-Based Encryption in Cloud Computing," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 983-993, 1 April-June 2021, doi: 10.1109/TETC.2019.2904637.

[25] H. Ma, R. Zhang, S. Sun, Z. Song and G. Tan, "Server-Aided Fine-Grained Access Control Mechanism with Robust Revocation in Cloud Computing," in IEEE Transactions on Services Computing, vol. 15, no. 1, pp. 164-173, 1 Jan.-Feb. 2022, doi: 10.1109/TSC.2019.2925028.

[26] C. Hahn, J. Kim, H. Kwon and J. Hur, "Efficient IoT Management With Resilience to Unauthorized Access to Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 1008-1020, 1 April-June 2022, doi: 10.1109/TCC.2020.2985046.

[27] H. Huang, Q. Tu, C. Jiang and M. Pan, "Nonsingular Terminal Sliding Mode Control Based on Sensor-Cloud System for Permanent Magnet in-Wheel Motor," in IEEE Access, vol. 8, pp. 140399-140410, 2020, doi: 10.1109/ACCESS.2020.3011922.

[28] J. -B. Lee, T. -H. Yoo, E. -H. Lee, B. -H. Hwang, S. -W. Ahn and C. -H. Cho, "High-Performance Software Load Balancer for Cloud-Native Architecture," in IEEE Access, vol. 9, pp. 123704-123716, 2021, doi: 10.1109/ACCESS.2021.3108801.

[29] S. T. Alshammari, K. Alsubhi, H. M. A. Aljahdali and A. M. Alghamdi, "Trust Management Systems in Cloud Services Environment: Taxonomy of Reputation Attacks and Defense Mechanisms," in IEEE Access, vol. 9, pp. 161488-161506, 2021, doi: 10.1109/ACCESS.2021.3132580.

[30] H. Wang, D. He and J. Han, "VOD-ADAC: Anonymous Distributed Fine-Grained Access Control Protocol with Verifiable Outsourced Decryption in Public Cloud," in IEEE Transactions on Services Computing, vol. 13, no. 3, pp. 572-583, 1 May-June 2020, doi: 10.1109/TSC.2017.2687459.

[31] S. J. De and S. Ruj, "Efficient Decentralized Attribute Based Access Control for Mobile Clouds," in IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 124-137, 1 Jan.-March 2020, doi: 10.1109/TCC.2017.2754255.

[32] K. Xue, N. Gai, J. Hong, D. S. L. Wei, P. Hong and N. Yu, "Efficient and Secure Attribute-Based Access Control With Identical Sub-Policies Frequently Used in Cloud Storage," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 1, pp. 635-646, 1 Jan.-Feb. 2022, doi: 10.1109/TDSC.2020.2987903.

[33] J. Hong et al., "TAFC: Time and Attribute Factors Combined Access Control for Time-Sensitive Data in Public Cloud," in IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 158-171, 1 Jan.-Feb. 2020, doi: 10.1109/TSC.2017.2682090.

[34] Z. Ghaffar, S. Ahmed, K. Mahmood, S. H. Islam, M. M. Hassan and G. Fortino, "An Improved Authentication Scheme for Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems," in IEEE Access, vol. 8, pp. 47144-47160, 2020, doi: 10.1109/ACCESS.2020.2977264.

[35] K. Edemacu, B. Jang and J. W. Kim, "Collaborative Ehealth Privacy and Security: An Access Control With Attribute Revocation Based on OBDD Access Structure," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 10, pp. 2960-2972, Oct. 2020, doi: 10.1109/JBHI.2020.2973713.

[36] R. Xu, J. Joshi and P. Krishnamurthy, "An Integrated Privacy Preserving Attribute-Based Access Control Framework Supporting Secure Deduplication," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 2, pp. 706-

721, 1 March-April 2021, doi: 10.1109/TDSC.2019.2946073. A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao and Y. Zhang, "A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5914-5925, 1 April1, 2021, doi: 10.1109/JIOT.2020.3032997.

[37] S. Xu, J. Ning, X. Huang, J. Zhou and R. H. Deng, "Server-Aided Bilateral Access Control for Secure Data Sharing With Dynamic User Groups," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4746-4761, 2021, doi: 10.1109/TIFS.2021.3113516.

[38] W. Susilo, P. Jiang, J. Lai, F. Guo, G. Yang and R. H. Deng, "Sanitizable Access Control System for Secure Cloud Storage Against Malicious Data Publishers," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 2138-2148, 1 May-June 2022, doi: 10.1109/TDSC.2021.3058132.

[39] Y. Hou, S. Garg, L. Hui, D. N. K. Jayakody, R. Jin and M. S. Hossain, "A Data Security Enhanced Access Control Mechanism in Mobile Edge Computing," in IEEE Access, vol. 8, pp. 136119-136130, 2020, doi: 10.1109/ACCESS.2020.3011477.

[40] R. Ghazal, A. K. Malik, N. Qadeer, B. Raza, A. R. Shahid and H. Alquhayz, "Intelligent Role-Based Access Control Model and Framework Using Semantic Business Roles in Multi-Domain Environments," in IEEE Access, vol. 8, pp. 12253-12267, 2020, doi: 10.1109/ACCESS.2020.2965333.

[41] X. Yang, R. Lu, J. Shao, X. Tang and A. A. Ghorbani, "Achieving Efficient Secure Deduplication With User-Defined Access Control in Cloud," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 1, pp. 591-606, 1 Jan.-Feb. 2022, doi: 10.1109/TDSC.2020.2987793.

[42] S. Xu, Y. Li, R. H. Deng, Y. Zhang, X. Luo and X. Liu, "Lightweight and Expressive Fine-Grained Access Control for Healthcare Internet-of-Things," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 474-490, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2936481.