Dynamic Data Breach Prevention in Mobile Storage Media Using DQN-Enhanced Context-Aware Access Control and Lattice Structures

Mounica Yenugula[1], Bhargavi Konda[2], Akhila Reddy Yadulla[3], Vinay Kumar Kasula[4]
[1,2,3,4]*Department of Information Technology, University of the Cumberlands, Williamsburg, KY, USA*
[1]*myenugula3188@ucumberlands.edu,* [2]*bkonda19519@ucumberlands.edu,*
[3]*ayadulla5882@ucumberlands.edu,* [4]*vkasula19501@ucumberlands.edu*

***Abstract*—**This study proposes an enhanced method for preventing data breaches in mobile storage media by improving access control mechanisms through the integration of Deep Q-Network (DQN) algorithms. Building on attribute-based encryption (ABE) with hidden ciphertext policies, a novel approach leveraging lattice structures for attribute policy descriptions is introduced. In this method, each attribute is represented as a lattice, and the set of attributes is structured into a product lattice. The DQN algorithm is employed to optimize dynamic decision-making in multi-level information flow control models, which govern access policies. This allows the system to adaptively refine access control, mitigating data breaches by learning from contextual information. The security and effectiveness of this new method are validated through theoretical analysis and experimentation. By incorporating DQN for dynamic learning and decision-making, the approach ensures fine-grained, context-aware access control in real time, especially in high-security environments. Furthermore, the integration of DQN simplifies policy management and enhances performance while maintaining robust security. The layered security management scheme designed for access control and breach prevention is tested, demonstrating both its flexibility and efficiency for securing sensitive information in pervasive computing environments.

Keywords*—Deep Q-Network, Attribute-Based Encryption, Mobile Storage Media, Lattice Security Model, Context-Aware Access Control, Data Breach Prevention*

I. INTRODUCTION

Mobile storage media—such as USB drives, external hard drives, memory cards, and iPods—are popular due to their compact size, high capacity, affordability, portability, and ease of use. They facilitate information exchange and can be used as bootable drives to create computing environments. Consequently, these devices have widespread applications across government, enterprise, and personal contexts. Despite their advantages, mobile storage media pose significant security risks. The potential for loss, theft, and misuse can lead to sensitive (confidential) information leaks, malware infections, and covert interactions with operating systems.

The issue of sensitive information leakage through mobile storage media is particularly pressing. While physical isolation technologies between internal and external networks theoretically create secure environments, the frequent data exchanges facilitated by these storage devices can inadvertently expose internal network information. For example, covert programs like Pod Slurping, which operate on devices such as iPods, can illegally download sensitive data from corporate systems, leading to substantial losses. The 2010 "WikiLeaks" incident, which involved the exposure of nearly 100,000 military documents related to the Iraq and Afghanistan wars, underscored the vulnerability of such systems. Investigations revealed that these leaks resulted from unauthorized copying of data onto mobile storage media by a U.S. military intelligence analyst stationed in Iraq. The Snowden affair in 2013 further emphasized the critical nature of sensitive information protection. Protecting sensitive information on mobile storage media is therefore a crucial area of research. This paper categorizes current research on mobile storage security into two main categories: 1) Security enhancement through integration, which involves embedding security controls directly into storage devices or dedicated hardware (e.g., incorporating smart card chips, dynamic isolation mechanisms, or security boards with USB interfaces and trusted execution environments); 2) Security enhancement through software, which focuses on improving the security control capabilities of trusted terminals for storage devices. This paper addresses the latter category.

Research on enhancing mobile storage media security can be divided into two aspects: pre-access authentication mechanisms and post-access control of sensitive information. Companies like DeviceLock, Beixin Source, and ZTE employ authentication mechanisms based on unique identifiers of mobile storage media, such as Vendor ID (VID), Product ID (PID), and Hardware Serial Number (HSN). Yang et al. proposed an authentication scheme for mobile storage media users based on the Schnorr protocol. However, both hardware-based unique identifiers and user authentication schemes still present certain security risks, as detailed in literature [16]. Lee et al. proposed a scheme leveraging the storage and computing capabilities of mobile devices to encrypt and store device identifiers and user identity information, initiating authentication and processing on the device end. This approach

is not suitable for storage media without computing capabilities and does not effectively control mobile device usage within internal networks. Literature [16] introduced a secure authentication protocol that binds mobile storage media to user identities, enabling effective management and restricted access within internal networks.

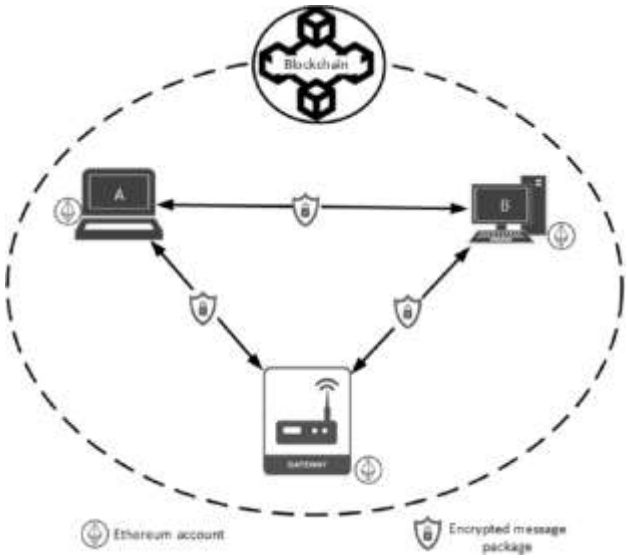

Figure 1: Layered Protection of Sensitive Information in Mobile Storage Media.

After authentication, protecting sensitive information on mobile storage devices primarily relies on encryption methods. Typically, symmetric encryption is used, where each authorized user is assigned a symmetric key. Upon successful authentication, the server provides the corresponding key, allowing the user to encrypt sensitive data before storing it on the mobile device or decrypt existing encrypted data. While symmetric encryption helps prevent information leakage, it presents two main security issues for mobile storage devices:

1) Inability to meet fine-grained access control requirements: Existing methods use a single symmetric key for all files copied by a user, which does not differentiate between files of varying sensitivity levels. This approach fails to address multi-level security needs. Even with centralized access control based on Public Key Infrastructure (PKI), users must obtain authorization certificates (keys) from an authentication server before accessing storage devices. This process can affect the efficiency of mobile storage media and compromise user privacy by listing identities.

2) Inability to adapt to dynamic access control requirements: Existing methods permit file encryption and decryption after successful mobile storage media authentication using static access control. Typically, a MAC address table for allowed host machines is created during system initialization. Policy changes require manual updates to this table, failing to dynamically control access permissions based on changes in users, access host security levels, or usage times. This static approach limits flexibility while attempting to ensure security.

To address these issues and meet the need for a flexible and convenient access control solution, a new approach is required. Deep Q-Networks (DQN), a type of deep reinforcement learning algorithm, offers a promising direction for enhancing access control in dynamic environments. By incorporating DQN, we can develop adaptive security solutions that learn and optimize access control policies based on real-time interactions and evolving contexts. DQN can be used to model and predict the security needs and access requirements dynamically. For example, DQN algorithms can be trained to recognize patterns in user behavior, device usage, and access contexts, allowing for fine-grained, context-aware security policies that adapt in real-time. This approach contrasts with traditional static methods by offering a more nuanced and adaptive access control mechanism, which can be compared to the methods discussed in literature [16], where static and hardware-based methods fall short in dynamic environments.

The integration of DQN into mobile storage security can provide a more flexible, efficient, and secure approach, addressing the limitations of existing encryption and access control methods while adapting to changing security contexts and user behaviors.

II. Attribute-Based Encryption Algorithm with Lattice Structures and Deep Q-Networks

A. Attribute-Based Encryption Algorithm

Attribute-Based Encryption (ABE) algorithms offer significant potential for fine-grained access control [21]. Unlike traditional public-key cryptosystems that rely on public key certificates, ABE allows encryption without knowing the exact identity of the decryption user. Instead, it uses a set of attributes as the public key and associates the ciphertext and user private keys with these attributes. Decryption is only possible if the decryption user's attributes match those required by the ciphertext. This approach eliminates the need for conventional public key certificates, greatly reducing the overhead associated with fine-grained data sharing and offering superior adaptability and flexibility compared to identity-based cryptographic mechanisms. The original basic ABE algorithm supported only threshold access control, where decryption is possible if the number of attributes intersecting with the ciphertext's attributes meets a predefined threshold. To represent more flexible access control policies, literature [22] introduced Key-Policy ABE (KP-ABE), where the decryption user defines the access policy, while literature [23] proposed Ciphertext-Policy ABE (CP-ABE), where the encryption user specifies the access policy for the ciphertext. These two algorithms cater to query-based and access control-based applications, respectively [21].

In CP-ABE algorithms, the access policy sent with the ciphertext is itself sensitive and requires protection. To prevent user attacks on the system, literature [24-26] proposed hidden access policy ABE algorithms using composite and prime groups. These algorithms hide some subset values during encryption to ensure that valid ciphertexts are distinguishable from invalid ones for authorized users. During decryption, users can determine if decryption is successful based on their attributes and the ciphertext, without needing to

know the exact access policy used during encryption. However, these methods require encryption users to list all possible attribute values, making the description of access policies lengthy and less understandable. If users omit or include incorrect attribute values, it can lead to access denial or leakage, rendering the access control system ineffective.

To address these challenges and enhance the practicality of hidden access policy ABE algorithms, this work focuses on introducing lattice structures into the access control policy. A new ABE algorithm is proposed that utilizes lattice structures, where each attribute forms a linear lattice or subset lattice, and the attribute set constructs a product lattice. This approach applies the previously proposed lattice-based multi-level information flow control model [27] to formulate access policies. The new mechanism maintains the advantages of existing hidden access policy ABE algorithms while simplifying the expression of access control policies, making it more suited for sharing sensitive information in multi-level security contexts and enabling fine-grained access control.

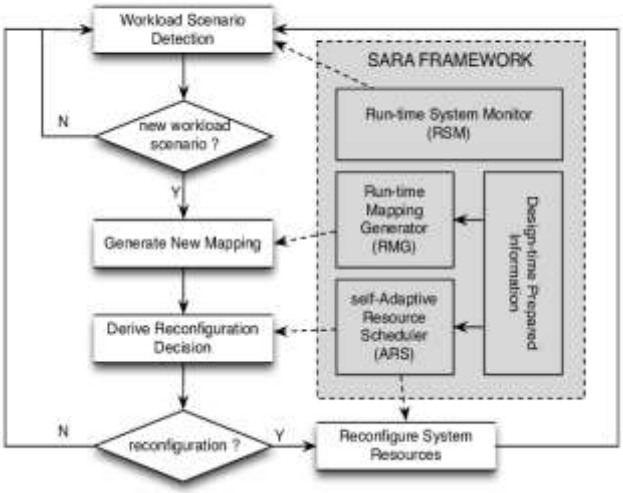

Figure 2: System Framework and Workflow

Integration with Deep Q-Networks (DQN)

Incorporating Deep Q-Networks (DQN) into this framework can enhance the dynamic adaptation of access control policies. By using DQN, the system can learn and optimize the access control strategies based on various environmental factors and user behaviors, improving adaptability and efficiency.

In comparison with traditional methods, integrating DQN allows the system to dynamically adjust access policies based on learned experiences from user interactions and environmental changes. This approach contrasts with static policy enforcement, which lacks flexibility in adapting to real-time changes. The use of DQN in this context could significantly improve the robustness and efficiency of access control mechanisms, addressing issues identified in the current literature and offering a more adaptive solution for managing sensitive information.

B. Improved CP-ABE Algorithm

1) Theoretical Foundations

Bilinear Mapping [28]

Let G_1 and G_2 be cyclic groups of prime order p, and let g be a generator of G_1. A mapping $e: G_1 \times G_1 \rightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

1. Bilinearity: For any $g, h \in G_1$ and $a, b \in Z\_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Computability: For any elements $g, h \in G_1$, there exists an efficient algorithm to compute $e(g, h)$.

Lagrange Interpolation [20], For $p_i \in Z$ and a set $S \subseteq Z_p$, the Lagrange coefficient is defined as

$$\Delta_{i(x)} = \prod_{\{j \in S, j \neq i\}} \frac{\{x - x_j\}}{\{x_i - x_j\}}$$

Given d points in S and the values $p(i)$ for each $i \in S$, the polynomial $p(x)$ of degree $d-1$ can be interpolated as

$$p(x) = \sum_{\{i \in S\}} p(i)\Delta_{i(x)}$$

Decisional Bilinear Diffie-Hellman (DBDH) Assumption [24], Given a security parameter k, assume that a challenger B can obtain a group G_1 and G_2 of prime order p, a bilinear map $e: G_1 \times G_1 \rightarrow G_2$, and a generator g of G_1. For randomly chosen $x, y, z \in Z_p$ and $Z \in G_2$, distinguishing Z from $e(g, g)^{xyz}$ is called the $(G1, G2, e)$ decisional bilinear Diffie-Hellman problem. If an attacker A solves the DBDH problem with advantage ϵ, then

$$\boldsymbol{Pr}[A(g^x, g^y, g^z, Z) = 1] - \boldsymbol{Pr}\left[A\left(g^x, g^y, g^z, e(g, g)^{\{xyz\}}\right) = 1\right] \geq \epsilon$$

Attribute Lattice Structure, Let $(A = \backslash \{A_1, A_2, \ldots, A_n\})$ be the set of all attributes, with n attributes in total. Each attribute A_i has n_i possible values, denoted as $V_i = \{v_{\{i1\}}, v_{\{i2\}}, \ldots, v_{\{in_i\}}\}$. Define a partial order $\leq$ on the values of any attribute A_i to describe the "less than or equal to" relationship between values. If $v_{\{i1\}\leq} v_{\{i2\}\leq} \ldots \leq v_{\{in_i\}}$, then $V_i, \leq$ is a linear lattice, and $(V, \leq)$ is a product lattice where $V = V_1 \times V_2 \times \cdots \times V_n$.

Access Policy, Let the attribute set of user U be $\{l_1, l_2, \ldots, l_n\} \subseteq V$, where $1 \leq i \leq n$. An access policy W is defined as $\{w_1, w_2, \ldots, w_n\}$, where $w_i \in V_i$. According to the lattice-based multi-level information flow control model, information can only flow from lower to higher security levels. Therefore, a file can only be decrypted if the user's attributes are at least as high as the ciphertext's attributes. Thus, if $U \leq W$ (i.e., $(l_i \leq w_i) \forall\, 1 \leq i \leq n)$, it is denoted that the user's attributes satisfy the access policy $UL \preccurlyeq W$.

2) Algorithm Description

The algorithm involves three entities: the authority center, the encryption user, and the decryption user. It consists of four main steps: initialization, encryption, key generation, and decryption.

Step 1: Initialization

The authority center generates system parameters $(G_1, G_2, p, g \in G_1, e)$, where G_1 and G_2 are cyclic groups of prime order p, and g is a generator of G_1. The bilinear map $e: G_1 \times G_1 \to G_2$ is also established. A random number $\alpha \in Z_p$ is selected, and $Y = e(g,g)^{\alpha}$ is computed. A set of random numbers $a_{\{i,t\}}$ corresponding to the attribute values $v_{\{i,t\}}$ is chosen, where $a_{\{i,t\}} \leftrightarrow v_{\{i,t\}}$. Define $g_{\{i,t\}} = g^{\{a_{\{i,t\}}\}}$, resulting in the public key $PK = \{Y, G_1, G_2, g, \{g_{\{i,t\}} | 1 \le i \le n, 1 \le t \le n\}\}$, the encryption key EK, and the master key $MK = \alpha$.

Step 2: Encryption

The encryption user uses the public key PK, encryption key EK, and access policy $W = \{w_1, w_2, \dots, w_n\}$ to encrypt plaintext M, resulting in ciphertext WM. Random number $r \in Z_p$ is selected, and $C_0 = g^r$ and $C_1 = M \cdot Y^r$ are computed. For each i and t, if $v_{i,t} = w_i$, set $C_{\{i,t\}} = g_{\{i,t\}}$. If $w_i < v_{\{i,t\}}$, then $C_{\{i,t\}} = E_{\{i,t\}}$; otherwise, $C_{\{i,t\}}$ is a random value $X_{\{i,t\}}$. The ciphertext is then $WM = \{C_0, C_1, \{C_{\{i,t\}} | 1 \le i \le n, 1 \le t \le n\}\}$.

Step 3: Key Generation

The authority center uses the master key MK and the decryption user's attribute set $L_U = \{l_1, l_2, \dots, l_n\}$ to generate the user's private key SKU. A random number $s \in Z_p$ is selected, and $D_0 = g^s$ and $D = e(g,g)^{\alpha s}$ are computed. For each i and t, if $v_{\{i,t\}} \le l_i$, then $D_{\{i,t\}} = g^{\{a_{\{i,t\}}\}}$; otherwise, $D_{\{i,t\}}$ is random. The private key is $SKU = \{D_0, \{D_{\{i,t\}} | 1 \le i \le n, 1 \le t \le n\}\}$.

Step 4: Decryption

The decryption user, using their attribute set \(L_U \) and private key \(SKU \), decrypts the ciphertext \(WM \) to obtain plaintext \(M \). For each \(i \) and \(t \), calculate \(t_j = \max(t) \) such that \(v_{i,t} \leq l_i \) and \(e(C_{i,t}, D_{i,t}) \neq E_{i,t} \). The decryption is successful if \(M = C_1 / \left(\prod_{i=1}^{n} e(C_{i,t}, D_{i,t}) \right) \).

3) Algorithm Analysis

- Correctness: When decrypting, the decryption user presents their attribute set $L_U = \{l_1, l_2, \dots, l_n\} \subseteq V$, and obtains their private key $SK_U = \langle D_0, \{D_{\{i,t\}}, E_{\{i,t\}} | 1 \le i \le n, 1 \le t \le n\} \rangle$. If the user's attribute set satisfies the access policy $W = \{w_1, w_2, \dots, w_n\}$, then there exists a subset $T \subseteq \{1, 2, \dots, n\}$ such that $w_i \in T$ for all i.

Since when $w_i < v_{\{i,t\}}$, $C_{\{i,t\}} = E_{\{i,t\}}$, it follows that in subset T, the values $v_{\{i,t\}}$ where $e(C_{\{i,t\}}, D_{\{i,t\}}) \neq E_{\{i,t\}}$ must equal w_i. Therefore, the correct decryption is achieved if

$$\prod_{\{i=1\}}^{\{n\}} e(C_{\{i,t\}}, D_{\{i,t\}}) = e(g,g)^{\{\alpha - s\}}$$

Thus, the decrypted message M is given by

$$M = \frac{C_1}{\prod_{\{i=1\}}^{\{n\}} e(C_{\{i,t\}}, D_{\{i,t\}})}$$

If the user's attributes do not satisfy the access policy W, then $w_i \neq T$, and some $C_{\{i,t\}}$ will be set to a random value $X_{\{i,t\}}$, making correct decryption impossible.

- Security

i. Security Model

The security model is defined using a game between the attacker A and the challenger B, as described in [24]. Preparation Phase: The attacker A submits two access control policies W_0 and W_1 to the challenger B. Initialization Phase: The challenger B initializes the system and generates the public key PK. First Phase: The attacker A queries the challenger B for the private key corresponding to the attribute set L_A. The challenger B returns the private key SK_A. In this phase, L_A must either satisfy W_0 or W_1, or not satisfy them, with the attacker able to make repeated queries. Challenge Phase: The attacker A submits two messages M_0 and M_1. The challenger B randomly selects $g_a \in \{0,1\}$, encrypts $M_{\{g_a\}}$ under the policy $W_{\{g_a\}}$, and returns the ciphertext to the attacker A. If L_A satisfies W_0 and W_1, then M_0 should equal M_1. Second Phase: The first phase is repeated. Guessing Phase: The attacker A guesses g'_a. The advantage of the attacker winning the game is given by $\boldsymbol{Pr}[g'_a = g_a] - \frac{1}{2}$. ABE algorithm with implicit lattice structure is considered secure if, for any polynomial-time adversary, the advantage of winning the above game is negligible.

ii. Security Proof

Assume the attacker A has a noticeable advantage ϵ in distinguishing between Game1 and Game0. We can construct a challenger B that solves the DBDH problem with advantage ϵ. The method is as follows: Preparation Phase: The attacker A submits two access control policies W_0 and W_1 to the challenger B. Initialization Phase: The challenger B sets up G_1, G_2, the bilinear map e, and the generator g. Randomly select $x, y, z \in Z\_p$ and $g_a \in \{0,1\}$:

- If $g_a = 0$, perform the game Game0 with parameters $(g, g^x, g^y, g^z, e(g,g)^{\{xyz\}})$.
- If $g_a = 1$, perform the game Game1 with parameters $(g, g^x, g^y, g^z, \delta)$, where δ is a random number.

Challenge: Randomly select $\alpha' \in Z_p$ \) and set $\alpha = xy + \alpha'$. Then $Y = e(g,g)$^{α'}. For each attribute A_i, randomly select values $k_{\{i,t\}}$ and set $g_{\{i,t\}} = g^{\{k_{\{i,t\}}\}}$. The public key PK is returned to the attacker A.

By analyzing the success probability of the attacker and the advantage of the challenger, we can show that the security of the ABE algorithm is guaranteed under the assumption of the hardness of the DBDH problem.

Phase 1: Key Query

1. Attacker's Key Query: The attacker A selects an attribute set L_A and queries for the private key. If $L_A \preccurlyeq W$ and $L_A \preccurlyeq A$,

then in the challenge phase, if $M_0 = M_1$, the attacker A has no advantage between Game0 and Game1. Therefore, we only need to consider the cases where $L_A \neq W$ and $A \neq L_A$.

Phase 2: Key Query and Guessing

1. Repeat Key Query: The attacker A performs another key query using the same method as in Phase 1.

2. Guessing: The attacker A outputs a guess ga' for the value of ga. If ga' = ga, the challenger B outputs 1; otherwise, it outputs 0. Based on the assumption, if the attacker guesses correctly in Game1 with probability ϵ better than in Game0, the constructed challenger B can solve the DBDH assumption with advantage ϵ.

Simplification of Access Policy

1.Comparison with Previous Work: Compared to previous works [24–26], the proposed algorithm simplifies the access policy, thereby reducing policy length. In previous literature, access policies were expressed using operators "and" for different attributes and "or" gates for different values of the same attribute, which could be cumbersome and increase policy length.

2. Improved Policy Representation: The proposed algorithm leverages the lattice-based multi-level information flow control model to represent attributes and their values. It uses a lattice structure to define the minimum values allowed for each attribute, thereby simplifying the policy.

3. Benefits: This approach maintains the advantages of traditional hidden access policy CP-ABE schemes while effectively simplifying the policy creation process. It reduces the complexity of encryption operations by minimizing the number of required exponentiations.

4.Comparison Table: Table 1 compares the proposed algorithm with the hidden access policy CP-ABE algorithms from [24–26], showing that the proposed method is more efficient in terms of policy length and complexity. The notation $|A|$ represents the total number of attributes, $|V|$ denotes the number of possible values for all attributes, and $|W|$ indicates the policy length, with $|V| \geq |W| \geq |A|$.

III. Context-Aware Access Control for Mobile Storage Devices

Mobile storage devices have numerous characteristics, such as a large number of stored resource files, uncertain and variable identities of resource users, and diverse usage environments. Traditional access control methods for mobile storage devices, which involve encrypting sensitive information after access authentication, are coarse-grained and static. These methods no longer meet the demands for security, convenience, and flexibility in mobile storage devices. Therefore, this paper proposes a context-aware access control approach and applies it to the security management of mobile storage devices.

A. Context-Aware Access Control Method

1) Context-Aware Access Control

The CP-ABE (Ciphertext-Policy Attribute-Based Encryption) algorithm disrupts the traditional notion in public-key encryption, where plaintext can only be decrypted by a unique private key. In CP-ABE, data creators encrypt data using an access control policy, and each data access requester has a decryption key corresponding to their attributes. As long as the requester's attributes match the access control policy, the key they possess can decrypt the data.

Before writing sensitive documents to a mobile storage device, the resource creator encrypts the document using an access control policy. The resource requester first generates a decryption key based on their attributes and then decrypts the ciphertext. Decryption will only succeed if the requester's attributes meet the access control policy, thus achieving access control over the resource. For example, if the resource creator defines an access control policy using a hierarchical structure, requiring that only managers or salespersons from Department 1 can decrypt the file, then requesters UA (attributes: {Department 1, Manager}) and UB (attributes: {Department 1, Salesperson}) can successfully decrypt the file. However, requester UC (attributes: {Department 2, Manager}) does not meet the access control policy and thus cannot decrypt the file. The key advantage of CP-ABE is that after plaintext encryption, multiple decryption keys that satisfy the access policy can be used for decryption. This makes CP-ABE suitable for applications in ubiquitous environments like mobile storage devices.

In CP-ABE applications, access policies typically use fixed attributes of the requester, such as organizational affiliation or identity, without considering the context during the access operation. To address the need for both security and flexibility in mobile storage devices, dynamic control of user access permissions based on different contexts is necessary. Therefore, this paper employs the attribute-based encryption algorithm with implicit lattice structures described in Section 2, using the context of mobile storage devices and users to construct access policies. After successful authentication, real-time attributes based on the current security context (such as user security level, accessing host security level, and access time) are obtained. This enables dynamic access control based on the user's security level, usage time, and environment, referred to as context-aware access control. This approach adapts access control dynamically based on real-time contexts without altering the access control policy, offering both security and high flexibility.

2) Lattices for Context in Mobile Storage Usage

To perform dynamic access control based on the current context of mobile storage usage, the context must be expressed as a set of attributes in a lattice structure.

Table 1: Comparison of Algorithms with References [24–26] (Worst-Case Scenario)

Operation	Reference [24]	Reference [25]	Reference [26]	This Study
Initial Stage				
Exponentiation	(2	V	+ 1)	(
Pairing	(2	W	+ 2)	(
Encryption				
Exponentiation	(3	A	+ 1)	(
Pairing	0	0	0	1
Key Generation				
Exponentiation	(3	A	+ 1)	(
Pairing	0	0	0	0
Decryption				
Exponentiation	(3	A	+ 1)	(
Pairing	0	0	0	0

Usage Context: The usage context of a mobile storage device consists of the user security level, host security level, and usage time security level, denoted as

$$\boldsymbol{Context} = \{(a_u, a_l, a_t) \mid a_u \in A_u \quad a_l \in A_l \; a_t \in A_t\}$$

where A_u is the set of user security levels, A_l is the set of host security levels, and A_t is the set of time security levels. The partial order $\leq$ on A_u represents the "less than or equal to" relationship, making $(A_u, \leq)$ a linear lattice, with $(A_l, \leq)$ and $(A_t, \leq)$ also being linear lattices. The usage context $(\boldsymbol{Context}, \leq)$ forms a product lattice, where $\boldsymbol{Context_1} \leq \boldsymbol{Context_2}$ if and only if $a_{\{u1\}} \leq a_{\{u2\}}$ *and* $a_{\{l1\}} \leq a_{\{l2\}}$ *and* $a_{\{t1\}} \leq a_{\{t2\}}$.

Based on the lattice-based multi-level information flow control model [27], sensitive information can only be accessed if the subject's security level is at least as high as the information's security level. Therefore, to prevent information leakage from the mobile storage device, the context of the access must satisfy which means that the security level of the context during file reading must be at least as high as the context level when the file was stored.

Table 2 shows the values for various context attributes. These context parameters are used as attributes in the implicit lattice structure ABE algorithm to control information access.

3) Information Access Control

To improve efficiency, this paper does not directly encrypt files using the proposed implicit lattice-based ABE algorithm. Instead, a symmetric key based on AES is generated for each file that needs to be encrypted. The file is first encrypted using Key, and then Key is encrypted using the implicit lattice-based ABE algorithm. Finally, the encrypted Key and the encrypted file are stored together as a single ciphertext on the mobile storage medium. During decryption, the implicit lattice-based ABE algorithm first decrypts to obtain Key, which is then used to decrypt the file.

B. Hierarchical Protection Scheme for Mobile Storage Media

This paper applies the principles of system security integrity and hierarchical protection to manage and control the use of external mobile storage devices, end-users, and trusted internal networks. This approach extends from controlling single-point security devices to managing security across the entire trusted domain and boundaries.

As illustrated in Figure 1, the use of mobile storage media is divided into three phases: pre-access, during access, and post-access. Multi-level security measures, including access authentication, health checks, permission management, and situational access control, are employed to protect sensitive information.

1. Pre-access Phase: This phase involves access authentication.
2. During-access Phase: After successful identity authentication, the mobile storage medium can connect to the host but cannot perform any operations yet. This phase focuses on health status detection, which includes checking both the connecting terminal and the mobile storage medium. This is crucial since mobile storage media can be infected with viruses when used externally, and similarly, viruses or malicious software on the connecting terminal can affect the security of the mobile storage medium.
3. Post-access Phase: Based on the health status results, appropriate permissions are assigned to the mobile storage medium, such as deny, read-only, or read-write access.

Once the mobile storage medium has the appropriate permissions, normal operations can proceed. Given that the primary function of mobile storage media is information access, controlling access to the information within is crucial for ensuring secure use. Files stored on the mobile storage medium are encrypted when being saved and can only be decrypted correctly when certain conditions are met.

The main workflow of the protection system is shown in Figure 2, involving five entities: Mobile Storage Medium (M), User (U), Host Terminal (C), Access Authentication Server (AS), and Authorization Server (PS). The access authentication server handles access authentication, the host terminal performs health checks, and the authorization server manages permission allocation.

The hierarchical protection workflow is as follows:

Step 1: The mobile storage medium requests connection to the terminal host, which reads the digital certificate from the mobile storage medium (1a) and receives the user's username and password (1b).

Step 2: The terminal submits the digital certificate, user information, and its own information to the authentication server.

Step 3: The authentication server performs access authentication based on the received information and returns the result to the terminal host (3a). If authentication is successful, the server forwards the attribute certificate request to the authorization server (3b).

Step 4: The terminal host performs access control on the mobile storage medium based on the authentication result.

Step 5: The authorization server issues a health status checker upon receiving the attribute certificate request.

Step 6: The host performs the required health check with the checker and sends the results to the authorization server.

Step 7: The authorization server assigns the appropriate permissions to the mobile storage medium based on the health check results.
Step 8: When the mobile storage medium performs specific file access operations, encryption and decryption operations are conducted.

1) Access Authentication

To address the security issues with current mobile storage access authentication mechanisms, a dual-factor authentication method based on binding the user to the storage medium is proposed. This method considers both user information and physical characteristics to effectively bind the user and the mobile storage medium. Detailed technical aspects are covered in reference [16] and will not be repeated here.

2) Health Status Detection and Permission Allocation

Health status detection involves a health checker issued by the authorization server, which assesses the health status of both the mobile storage medium and the host terminal.

Table 2: Contextual Attribute Values

Context Parameter	Value
User Security Level A_u	0 (Public), 1 (Secret), 2 (Confidential), 3 (Top Secret)
Host Security Level A_l	0 (Public), 1 (Secret), 2 (Confidential), 3 (Top Secret)
Usage Time Security Level A_t	0 (Leisure); 1 (Overtime), 2 (Work Hours)

C. Solution Analysis

The proposed scheme effectively manages access control for information on mobile storage media, demonstrating good security, flexibility, and efficiency. Here's an analysis of the solution with an emphasis on how it addresses different types of attacks, with additional insights on implementing Deep Q-Networks (DQN) for improved security:

1) Security

Mobile storage media face several types of attacks, including spoofing attacks, ferrying attacks, and abuse attacks [16]. The proposed solution's security features are analyzed with respect to these attack types.

2) Spoofing Attacks:

- Description: Spoofing attacks include impersonation (where a user gains unauthorized access by obtaining another user's storage device) and forgery (where an unauthorized device is used to mimic a legitimate one).
- Countermeasures: To combat spoofing attacks, the proposed solution employs a dual-factor authentication mechanism before the mobile storage media can be accessed. This method integrates the unique identifier of the storage media with user information. Only when a legitimate user connects a registered mobile storage device to the system can they obtain valid authentication. Unauthorized devices or users will be denied access. The technical details of dual-factor authentication are thoroughly discussed in reference [16].

3) Ferrying Attacks:

- Description: Ferrying attacks involve covertly transferring sensitive files from a trusted system to a mobile storage device using malicious software. Once the device is connected to an internet-connected computer, the malware sends these files to a specified destination.
- Countermeasures: To address ferrying attacks, the proposed solution incorporates dynamic authorization technology during the access phase. The mobile storage device undergoes a health check, and based on its health index, appropriate permissions are assigned. As described in Section 3.2.2, if a ferrying Trojan is present, the health index H(M) will be 1, resulting in a permission set $ACC(M) = DENY$. This prevents any read or write operations, thereby mitigating the risk of ferrying attacks.

3. Abuse Attacks:

- Description: Abuse attacks occur when a legitimate user connects a valid storage device but reads files under unauthorized conditions, leading to information leakage.
- Countermeasures: To mitigate abuse attacks, the proposed scheme uses the implicit lattice-based ABE algorithm for encryption and decryption based on different security contexts. This ensures that information does not leak from higher-security environments to lower-security ones, effectively preventing unauthorized access.

B. Implementation of Deep Q-Networks (DQN) for Enhanced Security

The proposed solution can be further enhanced by incorporating Deep Q-Networks (DQN) into the security framework. DQN can be used to optimize and dynamically adjust the access control policies based on evolving security threats and user behaviors. Here's how DQN can be integrated:

1) Dynamic Policy Adjustment:

- Description: Use DQN to train a model that learns optimal access control policies by interacting with the system and observing the outcomes of different access scenarios.
- Implementation: Train a DQN model to evaluate different access control strategies and adjust permissions dynamically based on real-time security data. For instance, if the DQN model detects an increased risk of a ferrying attack, it can adjust the health check parameters or access permissions to enhance security.

2) Adaptive Permission Management:

- Description: Utilize DQN to continuously learn and adapt the permission management system based on the patterns of legitimate and malicious activities.
- Implementation: Implement a DQN agent that monitors access logs and user behavior. The agent learns from these interactions to refine the permissions assigned to the mobile storage media, ensuring that permissions align with current threat levels and user behavior.

3) Real-Time Threat Response:

- Description: Deploy DQN to respond to real-time security threats by adjusting access controls and encryption/decryption protocols.

- Implementation: Integrate a DQN-based system that can detect anomalies or security threats in real-time. The system can then make immediate adjustments to access controls and encryption mechanisms to address these threats effectively.

To illustrate the security and application effectiveness of the proposed scheme, consider a simplified example:

During regular working hours, when user U connects a mobile storage medium M (with ID 43278) to a host machine C (with MAC address 44-45-53-54-00-00), and assuming authentication passes with read-write (RW) permissions, user U intends to copy a file named `abc.doc` from the host to M. The security levels of the user, host, and file attributes are as outlined in Tables 3 to 5.

The system's effective handling of security challenges ensures that only authorized operations are permitted, safeguarding sensitive information throughout the access and storage process.

C. Implementation with DQN Networking Algorithm

1) Request Handling

Entity C sends an operation request to PS with the following parameters: `Write|43278|abc.doc|44-45-53-54-00-00`. Based on the provided context values, PS determines the current context as `Context=(2,2,2)` and returns this result to C. C then uses the AES key `Key` to encrypt the file `abc.doc`. Next, C encrypts the key `Key` using the access policy `W=(2,2,2)` to produce `WK`. Both `WK` and the encrypted file are then stored together in the mobile storage medium (M).

2) Decryption Request

When user U (with ID=908) connects the storage medium (M) to another terminal host (C) with MAC address `F0-DE-F1-5F-18-5A` during a valid working period (assuming access authentication is successful and permissions are set to RW), U must send a read request: `Read|43278|908|F0-DE-F1-5F-18-5A` to PS. PS, using the provided information, determines the user's attributes `L=(2,1,2)` and calculates the decryption private key `SK`, which is then returned to U.

However, since the second attribute in `L` is less than the corresponding attribute in `W`, the implicit lattice-based ABE algorithm indicates that the user's attributes `L` do not meet the access policy `W`. Consequently, user U will not be able to correctly decrypt the key and, as a result, cannot retrieve the plaintext `abc.doc`. This mechanism ensures that information does not leak from higher security environments to lower security environments.

3) Flexibility

The proposed scheme employs contextual attributes such as user security level, usage time, and environment for encryption and decryption, allowing for fine-grained access control. This is more refined compared to the conventional approach where a single symmetric key is used uniformly for a user.

Furthermore, an essential function of mobile storage media is information exchange. In existing solutions, controlling the usage environment of the media typically involves backend binding, where a table of allowed host MAC addresses is established during system initialization. This table must be manually updated if policies change. In contrast, the proposed contextual access control method adapts dynamically to the real-time context of the user, ensuring high flexibility while maintaining security without altering the access control policy.

4) Efficiency

The implicit lattice-based ABE algorithm proposed in this paper simplifies the formulation of access control policies while maintaining the advantages of traditional hidden access policy ABE algorithms. It also reduces the computational complexity associated with encryption operations, as shown in Table 1. The new algorithm's design balances security with efficiency by employing a hybrid encryption scheme: symmetric encryption for files ensures high encryption performance, while implicit lattice-based ABE encryption of the symmetric key guarantees key security.

Table 6: Efficiency Comparison of Pure Symmetric Encryption and Hybrid Encryption with DQN Network (Unit: s)

File Size/KB	Pure Symmetric Encryption	Hybrid Encryption	DQN-Enhanced Hybrid Encryption
	Encryption	Decryption	Encryption
1,830	0.297	0.266	0.398
22,960	2.765	2.375	2.866
113,822	13.312	11.391	13.414
322,373	38.172	32.515	38.271

Table 6 compares the efficiency of traditional symmetric encryption solutions with the hybrid encryption scheme proposed in this paper. The experimental setup included a Lenovo PC with an Intel Core2 Duo E7500 2.93 GHz CPU and 2 GB of RAM. The symmetric encryption algorithm used was AES with a 256-bit key, and the encryption service classes were implemented in the .NET framework under the `System.Security.Cryptography` namespace. The results indicate that although the inclusion of implicit lattice-based ABE encryption introduces some additional processing time, the impact on efficiency is minimal. The trade-off is justified by the significant improvements in security and flexibility.

IV. CONCLUSION

Sensitive information leakage through mobile storage media is a significant issue today. This paper explores methods to enhance the security control capabilities of trusted terminals to effectively prevent sensitive information leaks from mobile storage devices.

The work presented in this paper focuses on two main areas:

1. Attribute-Based Encryption (ABE) Enhancement: Building upon the basic Ciphertext-Policy Attribute-Based Encryption (CP-ABE) method, we propose a novel approach that

integrates lattice-based structures for attribute policy descriptions. By representing the usage context of mobile storage media as lattice-based attributes, we leverage the inherent features of the ABE algorithm to achieve fine-grained and dynamic access control.

Table 3: User Security Levels

User ID	Security Level	Timestamp
908	2	1377151510
123	3	1376962830
–	–	–

Table 4: File Attributes

Filename	User ID	Timestamp
abc.doc	123	1378174815
–	–	–

Table 5: Host Security Levels

MAC Address	Security Level	Timestamp
F0-DE-F1-5F-18-5A	1	1373851233
44-45-53-54-00-00	2	1373934790

2. Dynamic Access Control through DQN: Traditional CP-ABE mechanisms, while flexible in representing access control policies and suitable for information sharing in distributed environments, suffer from verbose and less understandable policy descriptions. To address these issues, we propose a multi-level information flow control model based on lattice structures, which simplifies access control policy formulation while maintaining the benefits of hidden access policies. Furthermore, we incorporate Deep Q-Networks (DQN) to enhance the adaptability and efficiency of access control systems. DQN's capability to learn and adapt to dynamic environments allows for more effective and flexible management of access permissions based on real-time contexts such as user security levels, connected host security levels, and access times. The static access control methods currently employed for sensitive information on mobile storage devices do not adequately address the challenges posed by pervasive application environments. As user security levels and host security levels are often uncertain, a static approach is insufficient. The context-aware access control proposed in this paper, utilizing the novel lattice-based ABE algorithm and DQN, adapts access control dynamically based on the real-time context. This provides a balance between security and flexibility, ensuring that access permissions adjust in response to changing conditions while maintaining a high level of security.

The designed security management scheme for mobile storage media includes both pre-access authentication and post-access context-based control, offering multi-layered protection. This scheme integrates external mobile storage devices, terminal users, and trusted internal networks into a comprehensive security management framework. It extends from single-point security controls to a broader domain and boundary security control. The proposed methods are also applicable to sensitive information access control in pervasive application environments, providing a robust and adaptable solution to the challenges of mobile storage security.

REFERENCES

[1] IEEE Computer Society, "IEEE Standard for Authentication in Host Attachments of Transient Storage Devices," IEEE Std 1619.3-2010, 2010.

[2] Halsey, M., Beginning Windows 8, Apress, 2012.

[3] A. Tetmeyer and H. Saiedian, "Security threats and mitigating risk for USB devices," IEEE Technology and Society Magazine, vol. 29, no. 4, pp. 44-49, 2010.

[4] D. V. Pham, A. Syed, and M. N. Halgamuge, "Universal serial bus based software attacks and protection solutions," Digital Investigation, vol. 7, no. 3, pp. 172-184, 2011.

[5] GFI, "Pod Slurping-an Easy Technique for Stealing Data," GFI White Paper, 2011.

[6] H. Berghel, "WikiLeaks and the matter of private Manning," Computer, vol. 45, no. 3, pp. 70-73, 2012.

[7] S. Landau, "Making sense from Snowden: what's significant in the NSA surveillance revelations," IEEE Security & Privacy, vol. 11, no. 4, pp. 54-63, 2013.

[8] S. Z. Wu and C. Y. Shi, "Smart Card and USB Combined Equipment and Method for Communication with Computer," Chinese Patent 200710000328.3, 2007.

[9] J. Ma, Z. Y. Wang, J. C. Ren, et al., "TRSF: a positive protection framework for removable storage devices," Acta Electronica Sinica, vol. 40, no. 2, pp. 376-383, 2012.

[10] G. X. Zhang, C. Y. Shen, P. L. Wang, et al., "The research of dynamic track technology for removable storage information's trusted chain," Journal of Computer Research and Development, vol. 48, no. S1, pp. 37-42, 2011.

[11] DeviceLock, "Endpoint DLP suite," [Online]. Available: http://www.devicelock.com/dl/features.html, 2013.

[12] VRV SpecSEC, [Online]. Available: http://web.vrv.com.cn/products.html, 2013.

[13] ZTE Corporation, "Method to Implement Access Authentication, Equipment and a Mobile Terminal," Chinese Patent 200910133730.8, 2009.

[14] H. Q. Liao, J. Ling, Y. J. He, et al., "The research of unique identification for USB removable storage devices," Computer Engineering and Design, vol. 31, no. 12, pp. 2778-2780, 2010.

[15] F. Y. Yang, T. D. Wu, and S. H. Chiu, "A secure control protocol for USB mass storage devices," IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2339-2343, 2010.

[16] B. Chen, C. F. Qin, and L. Yu, "A secure access authentication scheme for removable storage media," Journal of Information & Computational Science, vol. 9, no. 15, pp. 4353-4363, 2012.

[17] K. Lee, K. Yim, and E. H. Spafford, "Reverse-safe authentication protocol for secure USB memories," Security and Communication Networks, vol. 5, no. 8, pp. 834-845, 2012.

[18] G. Z. Sun, D. W. Chen, D. R. Wu, et al., "The research and implementation of secure removable storage system," Computer Engineering, vol. 35, no. 11, pp. 116-119, 2009.

[19] SuperSpeed Semiconductors Co Ltd, "The Encryption of Removable Storage Devices Based on Synchronization of User and Master Authentication," Chinese Patent 201110184775.5, 2011.

[20] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2005), Aarhus, Denmark, 2005, pp. 457-473.

[21] J. S. Su, D. Cao, X. F. Wang, et al., "Attributes radical encryption mechanism," Journal of Software, vol. 22, no. 6, pp. 1299-1315, 2011.

[22] V. Goyal, O. Pandey, A. Sahai, et al., "Attribute-Based encryption for fine-grained access control of encrypted data," in Proceedings of the 2006 ACM Conference on Computer and Communications Security (CCS'06), Alexandria, VA, USA, 2006, pp. 89-98.

[23] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy attribute-based encryption," in Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 2007, pp. 321-334.

[24] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in Proceedings of the 6th International Conference on Applied Cryptography and Network Security (ACNS'08), New York, NY, USA, 2008, pp. 111-129.

[25] J. Lai, R. H. Deng, and Y. Li, "Fully secure ciphertext-policy hiding CP-ABE," in Proceedings of the 7th International Conference on Information Security Practice and Experience (ISPEC 2011), Guangzhou, China, 2011, pp. 24-39.

[26] J. Li, K. Ren, B. Zhu, et al., "Privacy-aware attribute-based encryption with user accountability," in Proceedings of the 12th International Conference on Information Security (ISC'09), Pisa, Italy, 2009, pp. 347-362.

[27] L. Yu, B. Chen, and J. M. Xiao, "The access control model of multiple strategies workflow management system," Systems Engineering - Theory & Practice, vol. 29, no. 2, pp. 151-158, 2009.

[28] D. Boneh and M. Franklin, "Identity-Based encryption from the Weil pairing," in Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2001), Santa Barbara, CA, USA, 2001, pp. 213-229.