ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240708001835
Original scientific paper

A Context Aware Security Model for Preventing Relay Attacks in NFC Enabled Mobile Devices

Davut CAVDAR*, Emrah TOMUR, Aysu Betin CAN

Abstract: Near Field Communication (NFC) is widely used in mobile applications, yet relay attacks remain a significant security risk, especially in access control systems. Existing countermeasures, such as distance bounding, ambient sensing, and RF fingerprinting, either lack adaptability or fail to provide comprehensive protection at the application layer. In this study, we propose M-CARBAC, a context-aware access control model designed to prevent relay attacks in NFC-enabled mobile devices. Unlike prior solutions, M-CARBAC integrates dynamic contextual verification, formal security definitions, and adaptive access control policies, ensuring that captured credentials remain invalid if relayed. A formal coverage analysis confirms that the model correctly evaluates all possible access requests. Performance tests conducted on an NFC-based access control testbed demonstrate that M-CARBAC effectively mitigates relay attacks while maintaining a reasonable computational overhead. Compared to existing solutions, our model offers a more robust and scalable approach for securing NFC transactions. These findings highlight M-CARBAC's potential for enhancing the security of mobile-based access control systems against evolving threats.

Keywords: access control; authentication; context aware; mobile device; near field communication (NFC)

1 INTRODUCTION

The popularity and usage volume of the smart mobile devices have exponentially increased in the last decade. The main reasons behind this situation are the functionality and convenience brought by mobile applications. Mark Weiser [15] described this age as the third era of computing in which electronic devices are smaller and able to interact with other devices. Within this era of computing, wireless communication technologies have started to be used more frequently in order to enhance interaction capabilities and data exchange between smart mobile devices. Various wireless communication technologies such as Wi-Fi, Bluetooth, IRDA, 4G/5G are used in a wide range of devices and solutions.

One such wireless communication technology, Near Field Communication (NFC), was introduced in 2002 by Philips and Nokia [14]. NFC was mainly emerged from mobility tendency and device interaction concept introduced in [15]. The main function of NFC is to establish connection between two mobile devices or NFC tags and reader. Data exchange or access requests can be easily performed using NFC-enabled smart mobile devices in daily applications creating smart solutions. Most popular usage of NFC technology is in contactless payments such as Google Wallet, loyalty couponing, ticketing in transport systems (bus, train etc.), gaming and smart gate systems.

Although used in critical and data sensitive solutions for quite a long period, security issues in NFC technology are not completely solved and standardized yet. NFC technology was introduced with RFID ISO/IEC 14443 "Contactless Proximity Smart Cards and their technical features" standard. It was first standardized as ISO/IEC 18092 "Near Field Communication Interface and Protocol (NFCIP - 1)" [11] and then ISO/IEC 21481 "Near Field Communication Interface and Protocol (NFCIP - 2)", but majority of NFC principles and functions are still inherited from RFID standards. Though NFC Forum has released even NDEF (NFC Data Exchange Format) and RTD (Record Type Definition) based standards, security requirements and standards are not discussed thoroughly. This situation causes some misconceptions in security-related issues for NFC-enabled mobile solutions.

For instance, NFC-enabled mobile devices have been used for long time in access control systems replacing passive cards, yet, in the literature, lots of research still work on finding solutions to prevent passive card relaying attack as given in Section 2. Because of their complexity in comparison with passive cards, mobile devices are exposed to security attacks more frequently; on the other hand, they are more powerful and flexible devices in terms of resources therefore they are more suitable environments to implement security solutions.

NFC devices interact with each other in one of three different modes depending on application types and needs. This flexibility creates a customizable environment for developers and manufacturers. In host card emulation mode, two active NFC devices, which are mobile device and reader, are used. Mobile devices act as the smart card based on ISO/IEC 14443 Type A, Type B and FelicCa standards in the application. When mobile phone gets close to reader, like RFID applications, reader initiates interaction and reaches mobile phone’s NFC module and Secure Element if available. Payment and access control applications use this mode for granting access. In Peer-to-Peer mode, two active NFC-enabled mobile phones interact with each other in two directional data exchange approach. First, one mobile device initiates communication setup process using request-response mechanism, then, other mobile device sends its response. In reader/writer mode, active mobile devices can both read and write passive NFC tags in Reader/Writer mode. Like RFID cards, NFC cards respond to requested data sent from initializer mobile device or write demanded data.

Use of host card emulation mode in applications also eases the practicability of relay attacks in this domain. Switching between Reader and Card Emulation modes causes relay attacks especially in NFC communication environments. Relay attack means requesting access using valid credentials by unauthorized users. Attackers first retrieve credentials during, before or after transactions, then forward them to another user. That receiver requests with credentials as real owner. NFC relay attacks have similarities in transaction flow; however, main difference is that mobile user does not need to have another user to complete attack. Mobile user can retrieve credentials in reader mode, then change its mode to host card emulation mode and request access using same device.

In our previous study [2], we also designed and developed a practical relay attack scenario to prove the applicability of such attacks in NFC environments. To prevent this proved security problem, we developed a strong precaution model including theoretical and practical applications. In this paper, we present this study, of which major contributions are as follows:

- A complete, dynamic, adaptive and context aware access control model to prevent relay attacks in NFC-enabled mobile applications.
- Formal definitions and descriptions of the proposed model are also presented to explain and prove the model theoretically.
- A practical implementation of the model is also developed in a realistic testbed where we implement an NFC-enabled door opening system.
- Performance of our model is tested in comparison to RBAC.
- Finally, we show that our model correctly responds to all possible access request combinations which can arise in a generic use case.

2 RELATED WORK

There are studies in the literature proposing countermeasures against NFC relay attack. Two of these countermeasures, distance bounding protocols and frame waiting time were offered in [5] and [7]. However, they could not provide sufficient protection against relay attack as explained below.

In the ISO/14443 standard, Frame Waiting Time (FWT) is defined for standard smart card and reader communication. FWT variable defines maximum response time after the end of the reader's data. Control of this variable was offered for avoiding relay attack in [5] and [7]. Attacker can modify this variable, or some additional readers may be placed between reader and card in order to overcome this variable. However, the main reason of its unsuitability is that different mobile devices are used in the proposed model instead of smart cards and when a relay attack occurs, only NFC-enabled mobile phones communicate with each other. There are no timing limitations between communications of two NFC-enabled mobile phones, because these devices are active, and it is different than the cases in the ISO/14443 based passive cards. Therefore, this problem needs to be solved with application layer security methods.

Location based control was also offered as another countermeasure in [7] and [6]. Distance-bounding protocols use FWT values for calculating round trip time and finally comment on requester's real location. As it was already mentioned, using FWT is not suitable solution for avoiding relay attack especially where NFC enabled mobile devices are used.

Drimer and Murdoch [4] offered a calculation of distance bounding security method for relay attacks based on smart cards, for Chip and Pin payment cards. Based on these tests, it measures round trip times according to low level signal transmissions and attempts to detect relay attacks.

Infrared Light was offered as a countermeasure for preventing relay attacks in mobile transactions in [8]. Proposed solution was implemented on six different test beds, but it was found that solution is strictly dependent on infrared sensor, i.e., hardware and therefore very hard to use in general. Chabbi et al. [3] presented an authentication protocol involving a server, a reader and an NFC-enabled smartphone capable of capturing and converting user's iris to a secret key. Also, they performed intrusion tests to inform smartphone's owner of attacks to determine the efficiency of the protocol. On the other hand, in this approach, mobile devices may not detect iris all the time because of lack of visibility and some additional processing time may be needed causing the operational problems.

Imran et al. [10] offered that Markov Chain may detect the relay attack on payment solutions using the principles of the chain, and the evaluation shows that in the case of electronic payment, Markov chain is successful in detecting anomalies in relay attacks. Consequently, they suggested that Markov chain algorithm could also be used as a protection against an attack in NFC payment. On the other hand, although Markov chain provides suitable solutions based on the trained data, it still works on estimation approach, there- fore it is risky to use it in the access control systems.

Anggoro et al. [1] proposed a method using symmetric cryptography to provide a more accurate detection protocol against threats in mobile NFC payment applications. The method was initially implemented on wireless short-range communication using Secure Element (SE) of mobile device. Creation of keys, encryption and management of certificates were performed in the SE. However, this approach addresses low-level operations in NFC transactions. Host Card Emulation (HCE) mode makes relay attacks possible at application level passing all controls in low-level.

Gurulian et al. [9] conducted an analysis using sensor data obtained from 17 sensors from a test platform for an emulated relay attack to determine whether they could effectively counteract these attacks. Each sensor, where possible, was used to record legitimate 350-400 and relay (il- legitimate) contactless transactions at two distinct physical sites. The research offered experimental results from which the effectiveness of ambient sensing can be assessed to provide a powerful anti-relay mechanism in applications that are sensitive to protection. Also, it is demonstrated that, under practical implementation environments, no single sensor evaluation is suitable for the security critical applications. In other words, raw sensor data should be analyzed and interpreted in a complete security model instead of standalone usage.

Li et al. [12] adapted the time-bound approach to detect relay transactions and present a quantitative estimate of normal transactions versus relayed transactions. A mobile prototype framework was also developed to demonstrate the feasibility of their proposed method.

Mousa and Dofe [16] proposed a hardware-encrypted PCB integrated with AES-128 encryption and accelerometer-generated dynamic keys. By capturing unique motion patterns to create unpredictable cryptographic keys, their approach mitigated 35 out of 37 simulated relay attacks. While this method demonstrates robustness in controlled environments, its reliance on specialized hardware components, such as secure element chips and accelerometer sensors, limits scalability in generic mobile ecosystems where software-driven, context-aware solutions are prioritized.

Abubaker and Gong [17] offered two protocols leveraging orthogonal frequency division multiplexing (OFDM) channel-fingerprinting to detect decode-and-forward (DF) and amplify-and-forward (AF) relay attacks. By analyzing the correlation of channel state information (CSI) and the distribution of channel frequency responses (CFR), their method detects relay attempts without requiring ultra-wideband hardware. While this approach achieves a low false-negative rate (FNR) of 12.5% in simulations and supports high data rates on existing hardware, it focuses primarily on physical-layer characteristics and lacks integration with dynamic context-aware mechanisms for application-layer security.

Recent work by Yang et al. [18] introduces NFC-RFAE, a semi-supervised RF authentication system for mobile NFC card systems. By combining supervised and unsupervised learning models, NFC-RFAE analyzes RF signal patterns (e.g., "ATQA" commands) to distinguish legitimate devices from malicious ones in multi-user scenarios like shared vehicle keys and mobile banking. The system achieves a 99% accuracy rate using random forests and anomaly detection (iNNE), demonstrating resilience against relay attacks without requiring labeled attack data. However, its reliance on RF signal analysis limits applicability in environments with high signal noise or hardware diversity, and it does not integrate dynamic context-aware mechanisms for real-time adaptation.

Wang et al. [19] proposed a deep-learning-aided RF fingerprinting method to detect NFC relay attacks at the physical layer by analyzing the unique waveform characteristics of transmitted signals. Their approach utilizes a convolutional neural network (CNN) to classify normal and relayed signals with high accuracy. They collected a dataset containing 66,366 NFC signal samples from both normal and relayed transactions and trained their CNN to distinguish between them. The experimental results demonstrated a detection accuracy of 99%, making RF fingerprinting a promising approach for relay attack mitigation. Unlike application-layer solutions, this method enables early detection of relay attacks before authentication interactions occur. However, its reliance on signal waveform features may limit its effectiveness in environments with high RF interference or hardware variations.

These proposed solutions are summarized in Tab. 1 and compared to our proposed method based on their main features. As shown, recent approaches like NFC-RFAE [18], Mousa and Dofe [16], Abubaker and Gong [17], and Wang et al. [19] tackle relay attacks using RF signal analysis, hardware encryption, or physical-layer defenses. While NFC-RFAE [18] achieves high detection accuracy using RF authentication, it lacks formal verification and dynamic policy adaptation. Similarly, Mousa and Dofe [16] and Abubaker and Gong [17] focus on hardware-level security, which is effective but not adaptable to dynamic NFC transactions. Wang et al. [19] introduced an RF fingerprinting-based deep-learning approach, detecting relay attacks with 99% accuracy, but it relies solely on signal waveform features without integrating context-aware security or access control policies.

Other studies, such as distance-bounding [5, 7, 12], ambient-sensing [9], and cryptographic protocols [1, 3, 10], provide partial countermeasures but fail to offer a holistic and adaptive security model. In contrast, M-CARBAC uniquely integrates context awareness, and dynamic adaptation, ensuring comprehensive protection at the application layer. Unlike RF-based or hardware-driven solutions, M-CARBAC prevents relay attacks even if credentials are compromised, making it a more robust and scalable security model for NFC-enabled systems.

3 MOBILE CONTEXT AWARE AND ROLE-BASED ACCESS CONTROL MODEL (M-CARBAC)

3.1 The Model Basics

We propose an access control model to prevent relay attack in NFC-enabled applications. Our proposed model is based on an extended version of role-based access control where we use context information to detect potential relay attacks. One of the main principles included in our proposal is that access credentials should be useless even when they are captured and be no longer valid even if they can be read by malicious parties.

Our proposed model provides application layer security mechanisms for NFC based access control systems. In addition to traditional and static security models, it includes principles of dynamic and adaptive context aware security approach.

3.2 M-CARBAC

The important part of our model is to ensure security using contextual information in NFC communication environment. Since relay attack occurs in application layer in cases where NFC-enabled mobile devices are used as both reader and smart card, Data Link Layer precautions such as calculating Frame Waiting Time for RFID based cards do not provide a complete solution. In our proposed context aware security model, context parameters are added and encrypted to the generated key in mobile device before sending. Thus, even if the data is relayed to another location or any device, it will not be valid, and this attack will not be successful. Our approach also validates previously set security principle stating that credential data should not be easily disclosed even it is captured and also it should be no longer valid even if it is deciphered.

Major features of the model are briefly described as follows:

- **Context Layer Abstraction**: Possibility of using different kinds of contextual information in authentication period.
- **Two Layer Encapsulation**: In addition to standard encryption, contexts are added to packet before sending access requests.
- **Runtime Creation**: Transactions and contextual information used in authentication are created in runtime to prevent relay attacks.
- **Non-descriptive Key Format**: In order to make sure that credentials will be useless, contextual information

and keys should not be easily identifiable by malicious parties.

- **Dynamic Adaptation**: Permission activation and revocations are performed in alignment with the context changes.

Table 1 The comparison of the main features of the proposed solutions

	M-CARBAC	[5, 7, 12]	[4]	[8]	[3]	[10]	[1]	[9]	[16, 17, 18]
NFC Transactions	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mobile Device Usage	✓	✓	-	✓	✓	-	-	✓	✓
Access Control Methodology	✓	-	-	-	-	-	-	-	-
Sensor Data Usage	✓	-	-	✓	✓	-	-	✓	-
Context Awareness	✓	-	-	-	-	-	-	-	-
Dynamic Adaptation	✓	-	-	-	-	-	-	-	-
Implementation & Test	✓	✓	✓	✓	✓	-	-	-	-
Formal Definitions	✓	-	-	-	-	-	-	-	-
Prevention of Relay Attacks in Application Layer	✓	-	-	-	-	-	-	-	-
Proposed as a Complete Security Model	✓	-	-	-	-	-	-	-	-

3.3 Principles and Components of the Model

Access control systems mainly regulate access requests of subjects on objects. When a user (subject) requests to perform an operation on a resource (object), access control mechanism evaluates this request based on access policies and predefined rules. Yet, these types of static access control mechanisms do not provide sufficient flexibility in dynamic mobile environments. Besides static credential information, contextual information is additionally used in context-aware models in order to provide flexibility in access control decisions.

We adopt such a context-aware approach in our proposed model and combine it with Role Based Access Control (RBAC). By doing this, we aim to both prevent relay attacks in mobile environments and also provide a dynamic access control for NFC-based applications.

In our model, credential key retrieved from a central server and additional contextual data are combined into a single contextual parameter before sending them to an authenticating entity in a non-descriptive and encrypted format. The user key is interpreted as static context and contextual data as dynamic contextual constraint by the system. User-Role assignments are performed based on static contexts and Role-Permission assignments, activation and revocation processes are performed according to dynamic contextual confirmation.

3.4 Access Control Mechanisms of the Model

Our proposed model is composed of several layers. These layers are Standardized Controls (SC), Context Sensitive Controls (CSC) and Dynamic Policy Controls (DPC) from bottom to top.

Our model aims to provide a security protection against relay attacks for physical access control systems which use smart mobile devices. We have designed a layered control mechanism for our model similar to the layered structure in computer networks in order to ensure a controlled flow in our method. At the bottom of layers, Standardized Controls take place. Above that, Context Sensitive Controls and Dynamic Policy Controls perform authentication and authorization processes respectively. Details of authentication and authorization processes in our model are explained in the following sections.

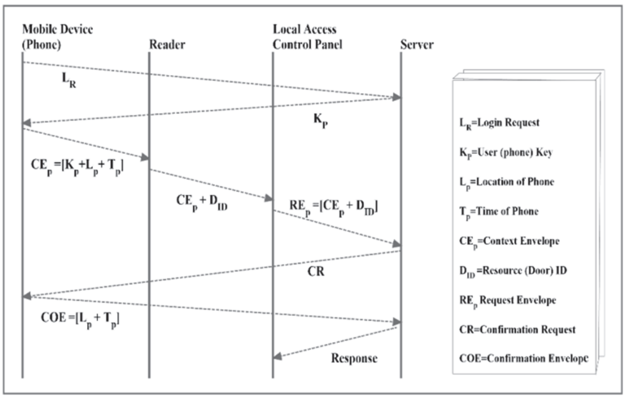

Figure 1 Conceptual flow in authentication process

3.4.1 Context Sensitive Controls (CSC)

In our proposed methodology, not only traditional static credentials but also dynamic context parameters are used for both role and permission assignments. Two-layer encapsulation is also applied to user key with context parameters. In this phase, static security model is extended with context parameters in order to provide dynamic control. Both user key and received context parameters are evaluated together. These operations are performed in the first cycle of controls.

In the second cycle, the verification of context parameters is performed by central server with the real sender phone in order to prevent relay attack. This verification is a crucial operation and our proposed model dictates that credential data should be useless even it is captured and also it should be no longer valid even if attackers read it. By doing this we ensure that even if user key is captured by an unauthorized party and relayed, system can detect this attempt and evaluate its validity and if not valid, finally blocks access to the requested resource.

The process of access control and authentication is depicted in Fig. 1. It firstly begins with mobile device user's login request (L_R) to the server. After logging in successfully, user-specific random key (K_P) generated by the server is sent to the user's mobile device. The generated key is valid for the duration specified by the user in the login process. At the end of this duration, the validation of the key is expired. The key obtained by the user who wants to access the resources, the time (T_s) and the location (L_p) information obtained automatically in background make so-called Context Envelope ($CE_p = [K_p + L_p + T_p]$). This envelope is sent to the reader. The reader adds to the envelope its own D_{ID} which describes the resource and sends them to the Access Control Panel. The Local Access Control Panel converts this information as the Request Envelope ($RE_p = [CE_p + D_{ID}]$) and sends it to the server. The server opens the packet arrived and controls the key (K_p), the time (T_p) and the location (L_p) information in the packet. If the key generated by the server is not expired, then the time (T_p) and the location (L_p) parameters are evaluated. If the request time is in the time intervals specified in the policies, the location parameter is then examined. Similarly, if the location information generated automatically in background by the mobile device is in the range specified in the policies, then the first control cycle is completed. The server either makes a decision or performs the second control cycle processes based on the authentication layer parameters. In this phase, the verification is performed in order to prevent relay attacks. The server sends the Confirmation Request (C_R) to the real mobile user logged and requests the location and the time information from the user in background. The real mobile user generates the Confirmation Envelope ($COE = [L_p + T_p]$) with the time and the location information calculated in background and sends it to the server. The server compares these time and location information with the ones arrived in the first cycle. The time spent for communicating is considered while comparing the time information. If the time and the location information are consistent with the ones arrived in the first cycle, and if they are also in the time interval specified in the policies, then positive, else negative response is sent to the Access Control Panel. The Access Control Panel either gives authorization or denies it depending on the response received.

In the second phase, a risk-based method can be used. These authorization mechanisms can be used when the application is critical, or when there is a user who has a suspicious process history, or when the location information is very close to the range limit specified. However, this method should be used for every request in situations that relay attack may occur. All these system level communication cycles are illustrated in Fig. 2.

In the relay attack scenario, even though the attacker impersonates the real user as it is close to the reader (when it is not in reality) by changing the L_p value in the message that attacker will send, request will be denied in the second phase of confirmation. If the attacker sends the location information of the real user in the Request Envelope in order to pass the second phase, the request will be denied because it is not close enough to the reader in this time. In this situation, the only condition that can surpass our proposed method is that the real user brings user mobile phone to a nearby place to the source and requests an access authorization.

In this situation, it can pass the location control in the second phase of confirmation. It is an exceptional and non-controllable situation that is very unlikely to happen in access control systems. In such a case, the software which runs on the real user's device can answer to the request sent by the server in the second phase only when it is run by the user actively. Thus, even though the attacker brings the mobile device of the real user to resource close enough, it cannot pass the second control phase.

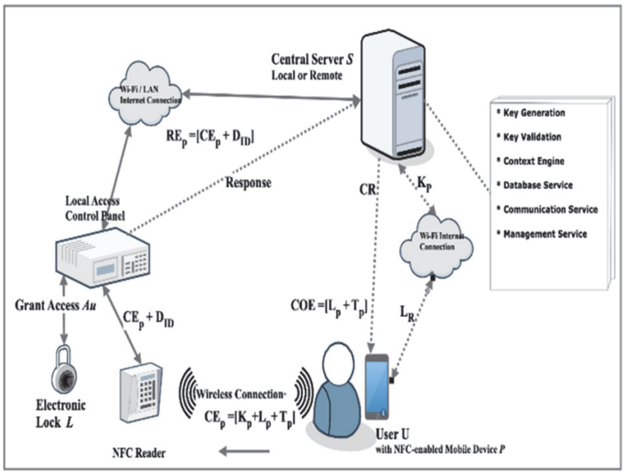

Figure 2 Credentials flow of system components

3.4.2 Dynamic Policy Controls (DPC)

After authentication process is performed successfully, authorization process is initiated with parameters coming from authentication phase. Also, both authentication and authorization transactions use system and context policies which are created based on application priorities and organization decisions.

Context parameters and user credential (key) are used both in authentication and authorization process but in different methods and different purposes. In the authentication phase, Context Sensitive Controls (CSC) tries to verify the identity of the requester, on the other hand, in the authorization period, these parameters are used for assigning verified requester to predefined roles and permissions with the help of policies. The key produced by server and, location and time contexts are extracted from Context Envelope ($CE_p = [K_p + L_p + T_p]$) in authentication period and then these parameters are evaluated in the upper authorization layer.

Also, as stated in Section 3.3, dynamic and static context information take part in role and permission assignments. In our model, location (L_p) and time of phone (T_p) are dynamic contexts and User Key (K_p) and Resource ID (D_{ID}) can be stated as static context. User-role assignments are performed based on static contexts; role-permission assignments are performed based on dynamic contexts. In other words, after authentication period, user based on (K_p) context is assigned to predefined role within the role hierarchy and whether per- mission is granted or not to requested resource is evaluated based on dynamic time and location contexts within predefined access policies. The methodology of dynamic policy controls is discussed in the following Formal Definitions section as well.

4 FORMAL DEFINITIONS

In this section, we give formal definition of our Mobile Context Aware and Role Based Access Control Model (M CAR-BAC). Our context aware model considers both static and dynamic context. We use *SMC* to denote the set of static mobile context information and *DMC* to denote the set of dynamic context information. A static mobile context $smc \in SMC$ denotes the static contextual information retrieved from a user before requesting access.

A dynamic mobile context information $dmc \in DMC$ denotes the information collected from internal and external domains. For example, ID of secure element of mobile device is an internal context whereas relative location, time and date are external dynamic context information.

M-CARBAC access control model formulated is $AC = (U, R, UR, PS, RP, SMC, DMC, O, Op)$ where U is the set of users, R is the set of roles defined in the system that are played by the users, $UR \subseteq U \times SMC \times R$ is the user-role assignment based on static context information, PS is the permission set, $RP \subseteq R \times DMC \times PS$ is the role-permission assignment based on dynamic context information, O is the set of objects that user wants to access such as door, and Op is the set of operations defined on objects such as activate. Note that, the user-role assignment UR is different from RBAC as it contains static context as well. Also, the role-permission assignment RP is different from RBAC since it contains dynamic context information such as location and time.

Given a user $u \in U$ and a static context $smc \subseteq SMC$, the set of roles played by u in this context in AC is retrieved by the function getUserRole(u, sc) = $r \in R|(u, sc, r) \in UR$.

The permission set $PS \subseteq O \times Op \times$ Powerset(allow.deny) describes system permission on objects and operations. Given $p \in PS$ where $p = (ob\ j, op, type)$, we use operator . (dot) to represent the parts of p such as $p.ob\ j$ denotes the object and $p.op$ denotes operation that permission is defined. Given a role $r \in R$ and dynamic context information $dmc \subseteq DMC$, the set of permissions in AC is retrieved by the function get RolePermissions(r, dmc, $ob\ j$, op) = $\{p \in PS|(r, dmc, p) \in RP \wedge p.ob\ j = ob\ j \wedge p.op = op\}$. Finally, we define evaluate-Permission function that takes a set of permission rules and returns the union of all their response sets, i.e., evaluate Permission(P) = $U(p \in P)$p.type where $P \subseteq PS$.

In our model an access is granted if all the following conditions, where C_i represents condition, are satisfied:

- C_1: The user within its current static context is assigned to a role.
- C_2: All the roles of the user have permission to access the object to perform the requested operation in the current context. That is, for each role, there is at least one permission rule with allow and no permission with deny on the request.
- C_3: The time and location context in the confirmation envelop (COE) in the second cycle should match with the time and location in the dynamic context of the request to prevent relay attacks. This condition states the main contribution of the model. Instead of the evaluation of the context that is provided in the request envelope, the model validates the confirmed context parameters in the background as second cycle of transactions to prevent relay attack. In addition, custom context verification rules, such as proximity of the object to the requester and time context being within the time interval of the permission, must hold.
- C_4: Risk assessment on the role and permission does not fail. While the model does not prescribe a specific technique, it requires a risk assessment in the evaluation process.

First, we define expressions to check these conditions, then we give the formal expression of our model. The first condition C_1 is expressed as get UserRole(u, smc) = $R \wedge R = \emptyset$.

Let $r \in R$ be a role assigned to user that is requesting access on object $ob\ j \in O$ to perform operation $op \in Op$, and let $dmc \subseteq DMC$ be the dynamic context information of the request. Checking this request against the policy setting is policy Eval (r, dc, $ob\ j$, dmc) = get Role Permissions (r, dc, ob j, op) = $Pi \wedge Pi = \emptyset \wedge$ evaluate Permissions (Pi) = {allow}. The first part of the conjunction collects all the permission rules declared on obj and op associated with the role r within the dynamic context defined in the role-permission assignment RP of the access control AC. The second part of the conjunction ensures that there is a permission rule associated to the role r. The third term uses evaluate Permissions function which computes the union of the last part of the permissions in Pi. There are four possible outcomes of this function: $\emptyset$, {allow}, {deny}, {allow, deny}. Having a constraint that evaluate Permission (Pi) = {allow} ensures that there is at least one allow and there are no deny permissions. This statement uses the µdeny overrides the allow" principle for the model. Not only in NFC, but also in wireless or other network communication, deny rules always overrides the allow rules. The condition C_2 is expressed with the policy Eval function.

The condition C_3 is context verification. Given context information $dmc \subseteq DMC$, let L_p be the location of the requester device extracted from dmc and let T_p be the current time of the requester device in dmc. Also, as a contribution of proposed model, confirmed contextual information CL_p (Confirmed location value) and CT_p (Confirmed Time Value) are retrieved from context of Confirmation Envelope (COE) that is executed in the background at the second cycle control of the model (see Section 3). We define verifyContext(dmc, smc) = checkRelay(dmc) $\wedge$ checkContext(dmc, smc) where checkRelay(dmc) = $(L_p = CL_p) \wedge (T_p = CT_p)$. The first part aims to prevent relay attacks. It ensures that confirmed context values, which are retrieved in the second cycle of controls, should match with the contextual information of the requester device. If L_p is not equal to CL_p, there may be a relay attack attempt because the time value of requester and confirmed value from authenticated user are different. A similar argument is valid for the time context. The model validates the confirmed context parameters in the background in second cycle of transactions to prevent relay attack instead of evaluating the context provided in the request envelope. The second part of context verification, denoted as checkContext, can vary for different system requirements. System designers may employ different dynamic context elements with different control mechanisms such as using biometric values. Here we give a sample context validation as checkContext(dmc, smc) = $(|x - L_p| \leq \Theta) \wedge (T_s \leq T_p \leq T_e)$ where x is the location of the object Θ is the threshold distance value for evaluation, T_s and T_e are the start and end times of access rule. Location of requester device (L_p) is expected to locate within the predefined threshold distance value (T_d) and current time value of the requester device (T_p) should be between start and end time of related predefined rule.

The condition C_4 is expressed with $evalRisk$ function.

It represents risk evaluation calculated dynamically at run time. Assessment can be made based on suspicious transaction history, different forms of behavior or previously de- fined risk factors. Also, AI algorithms and machine learning principles can take place in risk evaluation.

M-CARBAC final evaluation of an access request is as follows. The access control system *AC* = *(U*, *R*, *UR*, *PS*, *RP*, *SMC*, *DMC*, *O*, *Op)* receives a request by a user *u* on object *obj* to perform *op* with the context information *smc* ⊆ *SMC* and *dmc* ⊆ *DMC*. The request is granted using the following expression if and only if the following expression is true.

$$\begin{aligned}&grantAccess(u,smc,dmc,o,op) \Rightarrow (smc \neq \emptyset \land dmc \neq \emptyset)\land\\&getUserRole(u, smc)=R \land R\neq\emptyset\land \ (\forall r \in R,\\&getRolePermissions(r, dc, ob\,j, op)=Pi\land\\&Pi\neq\emptyset \land evaluatePermissions(Pi) = \{allow\})\land\\&verifyContext(dmc, smc)\land evalRisk(r, ob\,j, op, dmc, smc)\end{aligned} \tag{1}$$

5 IMPLEMENTATION AND TEST

5.1 NFC Access Control System

The implementation of infrastructure and framework of the model is developed in order to apply and verify the model practically. This complete system infrastructure consists of two types of structure which are hardware and software modules. Hardware part includes Central Server, Local Access Control Panel, NFC Reader and Mobile Device. Other part consists of development of central software and database, mobile application, wireless communication structure based on web service basics.

Majority of system transactions are performed by central server software, which is designed to be modular. The software modules are Database Service, Communication Service, Session Manager, Role Manager, Permission Manager, Context Engine, Conflict Engine and Key Engine. After logging onto system, key engine generates user key and sends it through communication service based on web service. User creates request package with his key and contextual information and sends it through NFC reader when he wants to access resource. Context engine interprets and verifies the retrieved context in the package in its submodules. According to predefined rules, conditions and tuples that are retrieved from database service, required assignments are sent to role and permission manager. Also, according to mentioned conditions, context engine may revoke these assignments when needed. Finally, response is sent to both physical access control component and mobile device again through communication service in wired connection and web service. The modules and transactions between them are illustrated in Fig. 3.

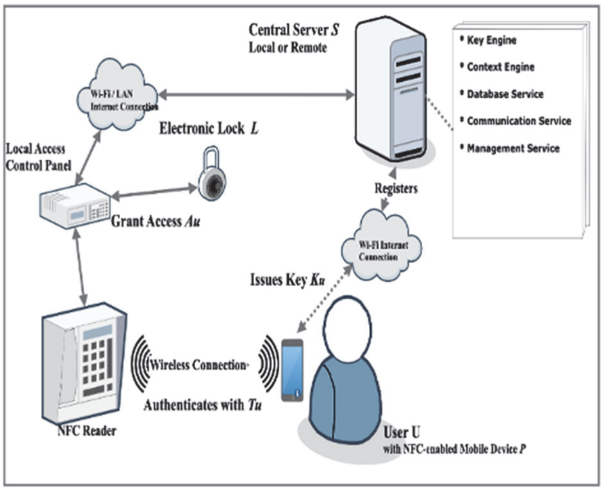

Figure 3 NFC based access control infrastructure for proposed model

5.2 High-Level Use Cases of the Implementation of the Model

Although variety of scenarios can be implemented in the testbed to show validity and coverage of the model, the high-level use cases can be categorized into four main groups.

The first group includes initial checks, second group is Relay User which includes relay attack attempts using contextual parameters. Because proposed model aims to detect and prevent the Relay Attacks in the proposed scope, this use case group is studied and implemented in order to show the validity of the model. The other two groups include the access trials of the normal users. The proposed model is developed at the top of the Role Based Access Control Model (RBAC), therefore the main principles of the RBAC also should be verified in the implementation. Based on the principles of the RBAC and the proposed model, requests of the normal users are evaluated using their contextual information like Relay Users and "Deny" and "Allow" responses are generated after applying policy rules. These high level uses cases are provided in Tab. 2.

Table 2 High level use cases of the implementation

Group	Use Case Identifier	Description of Use Case	Formal Representation	Response
Initial Checks	IC1	No contextual information.	c == Null	Deny
	IC2	Subject is not assigned to any role.	$getUserRole(s,\prod_{smc}c)$==Ø	Deny
	IC3	No rule is defined.	$getRolePermissions(r,\prod_{smc}c,\prod_{obj,op}p)$==Ø	Deny
Relay User (Attack)	RU1	The Relay Attackuse case (location).	Lp≠ CLp	Deny
	RU2	The Relay Attackuse case(time).	Tp≠ CTp	Deny
Normal User (Deny)	NUD1	Location context is not verified.	$L_p=CL_p\land T_p=CT_p\land\lvert x-L_p\rvert>T_d$	Deny
	NUD2	Time context is not verified (before).	$L_p=CL_p\land T_p=CT_p\land\lvert x-L_p\rvert \leq T_d\land T_p<T_s$	Deny
	NUD3	Time context is not verified (after).	$L_p=CL_p\land T_p=CT_p\land\lvert x-L_p\rvert\leq T_d\land T_p>T_e$	Deny
	NUD4	Deny result exists.	$\exists q\in R(q=Deny)$	Deny
Normal User (Allow)	NUA1	All checks are verified.	$L_p=CL_p\land T_p=CT_p\land\lvert x-L_p\rvert\leq T_d\land T_s\leq T_p\leq T_e$	Allow

5.3 Coverage Analysis of the Model

Based on the high-level use cases of the model above, we have analyzed the coverage of the model in terms of possible request combinations. In order to demonstrate all request combinations, attributes are identified and grouped as categories. The sets of categories are as follows:

Role Category (RoC)

1. No Role - *getUserRole*(s, $\prod_{smc} c$) = =Ø
2. Role(s) Exist - Dependent to other categories (RuC, CoC)

Rule Category (RuC)

1. No Rule - *getRolePermissions*(r,$\prod_{dmc} c$,$\prod_{obj,op} p$) = = Ø
2. Rule(s) Exist - At Least One Deny - $\exists q \in R(q = Deny)$
3. Rule(s) Exist - No Deny - At least One Allow - Dependent to other categories (CoC)

Context Category (CoC)

1. Context not provided (null)
2. Not Confirmed Location of Phone - Clp_i
3. Not Confirmed Time of Phone - Ctp_i
4. Relay Context Confirmed but Invalid Location of Phone Context - Lp_i
5. Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, $T\,p_i$
6. All Contexts are confirmed - CLp_v,CTp_v, Lp_v, Tp_v

$CoC = \{null, \{(CLP_i,c)|c \in NUC\}, \{(CTP_i,c)|c \in NUC\}, \{CLP_v,CTP_v,Lp_i\}, \{CLP_v,CTp_v,Lp_v,Tp_i\}, \{CLp_v,CTp_v,Lp_v,TP_v\}$

where: Lp_i - Not Confirmed Location of Phone Context; Lp_v - Confirmed Location of Phone Context; Tp_i - Not Confirmed Time of Phone Context; Tp_v - Confirmed Time of Phone Context.

Finally, all request combinations are the cartesian product given as: $RoC \times RuC \times CoC = \{(a, b, c)|a \in RoC$, b ∈RuC and c ∈ CoC}, the number of the elements in the cartesian product is: $s(RoC \times RuC \times CoC) = s(RoC) * s(RuC) * s(CoC) = 2 * 3 * 6 = 36$ elements. These 36 request combinations are all covered by 4 groups of high-level use cases described in this chapter. According to initial checks, when context is not provided, role or rule is not defined then implementation does not evaluate other parameters in the requests. Based on this approach, one high-level use case can cover many combinations. All the combinations and high- level use case which cover these combinations are illustrated in Tab. 3.

Table 3 All possible combinations of the requests

	Role Category (RoC)	Rule Category (RuC)	Context Category (CoC)	High Level Use Case
1	No Role (RoC1)	No Rule (RuC1)	Context not provided (CoC1)	IC1
2	No Role (RoC1)	No Rule (RuC1)	Not Confirmed Location of Phone - Clp_i - (CoC2)	IC2
3	No Role (RoC1)	No Rule (RuC1)	Not Confirmed Time of Phone - Ctp_i- (CoC3)	IC2
4	No Role (RoC1)	No Rule (RuC1)	Relay Context Confirmed but Invalid Location of Phone Context - Lp_i - (CoC4)	IC2
5	No Role (RoC1)	No Rule (RuC1)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	IC2
6	No Role (RoC1)	No Rule (RuC1)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	IC2
7	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	Context not provided (CoC1)	IC1
8	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	Not Confirmed Location of Phone - Clp_i - (CoC2)	IC2
9	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	Not Confirmed Time of Phone - Ctp_i - (CoC3)	IC2
10	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	Relay Context Confirmed but Invalid Location of Phone Context - Lp_i - (CoC4) - (CoC3)	IC2
11	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	IC2
12	No Role (RoC1)	Rule(s) Exist - At Least One Deny (RuC2)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	IC2
13	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Context not provided (CoC1)	IC1
14	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Not Confirmed Location of Phone - Clp_i - (CoC2)	IC2
15	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Not Confirmed Time of Phone - Ctp_i - (CoC3)	IC2
16	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Relay Context Confirmed but Invalid Location of Phone Context - Lpi - (CoC4) - (CoC3)	IC2
17	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	IC2
18	No Role (RoC1)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	IC2
19	Role(s) Exist (RoC2)	No Rule (RuC1)	Context not provided (CoC1)	IC3
20	Role(s) Exist (RoC2)	No Rule (RuC1)	Not Confirmed Location of Phone - Clp_i - (CoC2)	IC3
21	Role(s) Exist (RoC2)	No Rule (RuC1)	Not Confirmed Time of Phone - Ctp_i - (CoC3)	IC3
22	Role(s) Exist (RoC2)	No Rule (RuC1)	Relay Context Confirmed but Invalid Location of Phone Context – Lpi - (CoC4) - (CoC3)	IC3
23	Role(s) Exist (RoC2)	No Rule (RuC1)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	IC3
24	Role(s) Exist (RoC2)	No Rule (RuC1)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	IC3
25	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	Context not provided (CoC1)	IC1
26	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	Not Confirmed Location of Phone - Clp_i - (CoC2)	NUD4
27	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	Not Confirmed Time of Phone - Ctp_i - (CoC3)	NUD4

Table 3 All possible combinations of the requests (continuation)

	Role Category (RoC)	Rule Category (RuC)	Context Category (CoC)	High Level Use Case
28	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	Relay Context Confirmed but Invalid Location of Phone Context - Lpi - (CoC4) - (CoC3)	NUD4
29	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	NUD4
30	Role(s) Exist (RoC2)	Rule(s) Exist - At Least One Deny (RuC2)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	NUD4
31	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Context not provided (CoC1)	IC1
32	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Not Confirmed Location of Phone - Clp_i- (CoC2)	RU1
33	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Not Confirmed Time of Phone - Ctp_i- (CoC3)	RU2
34	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Relay Context Confirmed but Invalid Location of Phone Context – Lpi - (CoC4) - (CoC3)	NUD1
35	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	Relay Context Confirmed & Valid Location but Time of Phone Context not Confirmed - Lp_v, Tp_i - (CoC5)	NUD2, NUD3
36	Role(s) Exist (RoC2)	Rule(s) Exist - No Deny - At least One Allow (RuC3)	All Contexts are confirmed - CLp_v, CTp_v, Lp_v, Tp_v - (CoC6)	NUA1

5.4 Discussion and Analysis

The dual-phase authentication mechanism in M-CARBAC enhances security by validating contextual parameters at runtime; however, this real-time processing requirement may introduce slight delays in environments with high NFC transaction volumes, particularly in large-scale access control systems. Additionally, the model is specifically designed for NFC-based authentication, making it less adaptable to other wireless communication protocols such as Bluetooth or Wi-Fi without significant modifications. While M-CARBAC strengthens security through context verification and risk assessment, these additional authentication layers may impact user experience, requiring extra interactions or background computations that could slightly delay access authorization. In scenarios where users expect instant access, such as high-speed transportation systems or time-sensitive transactions, the trade-off between security and usability must be carefully considered.

We have carried out performance tests to evaluate model's real time metrics in our test bed. The results may change depending on environmental parameters such as hardware and software, programming languages or database types. To avoid these effects, we performed this performance test in identical environments.

Two separate test sets have been used to compare performance of response times of the proposed model. First test set has a request composition of mixed types of different high-level use cases. This composition is formed as 10% IC1 - 10% IC2 - 10% IC3 - 15% NUD1 - 15% NUD2 - 15% NUD3 - 15% NUD4- 15% NUA1. The second set composition consists of 50% RU1- 50% RU2 requests.

We aim to analyze the performance of the model for both mixed compositions of different types of access requests and relay requests.

The results of first test set which includes mixed types of use case requests are illustrated in Fig. 4 and the results of second test set which includes mixed relay attack use case requests are illustrated in Fig. 5 as line graphs in terms of number of access requests and times required for these evaluation operations based on the principals of the models RBAC and the proposed model.

The performance of our model is reasonable and a bit lower than RBAC model as expected due to the overheads of using context Information.

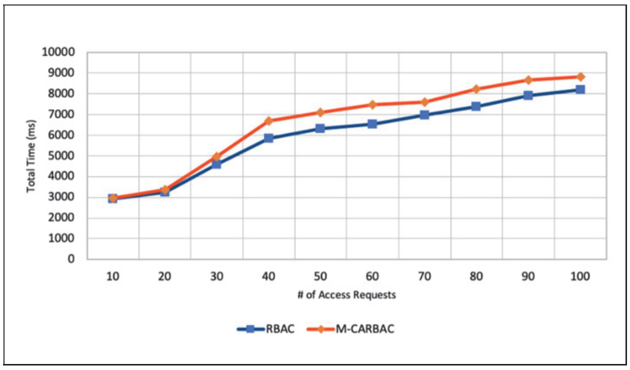

Figure 4 Comparison response times of RBAC and M-CARBAC (mixed use cases)

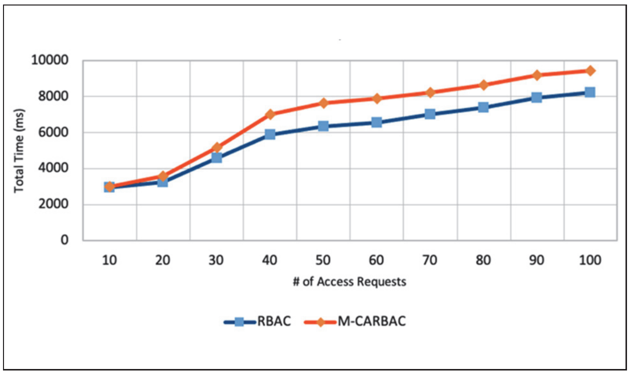

Figure 5 Comparison response times of RBAC and M-CARBAC (relay attacks)

According to the study in [13], the performance of RBAC is better than all its derivatives like our model. The increase in the number of queries results in increase of the response time of both models as natural. According to the results, a little bit more time is needed for requests containing only relay attack than those containing a mixed request composition because the relay attack requests are evaluated in both first and second cycle of the controls. On the other hand, some user requests can be denied in the Initial Control (IC) period such as requests including no context etc. before further evaluations.

In addition to test response and process times of queries, we have also performed and analyzed increase in CPU usage comparing RBAC, which our model is based on, and M-CARBAC. These CPU usage tests are performed on the completely same hardware configuration with 25, 50, 75 and 100 queries using first test set which includes requests of mixed-use cases. Then, average CPU usages during request handling are calculated automatically based on the data retrieved from benchmark of system hardware. The results of the CPU usage of models are illustrated in Fig. 6 as a line graph. According to the results, our model needs more CPU power for all numbers of requests than RBAC. For the maximum case, our model needs 3% more CPU. This difference is a relatively acceptable difference because it is relatively small and our model performs more evaluations and controls with evaluation functions; therefore, they need some extra CPU source.

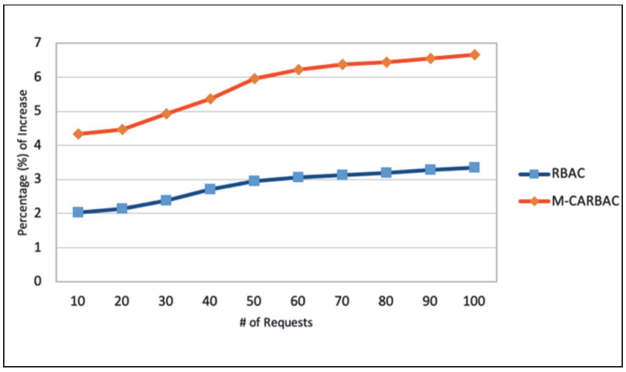

Figure 6 Comparison of increase in CPU usage of RBAC and M- CARBAC (mixed use cases)

The experimental results demonstrate that M-CARBAC successfully prevents relay attacks ensuring that unauthorized access attempts are blocked even when authentication credentials are compromised. This is achieved through context-aware verification, which dynamically evaluates location and time constraints before granting access. Compared to traditional RBAC, M-CARBAC adds an additional verification layer, making it significantly more resistant to relay attacks.

6 CONCLUSION

This study introduced M-CARBAC, a novel context-aware access control model aimed at preventing NFC relay attacks. M-CARBAC ensures that authentication credentials remain protected from exploitation in relay attacks through the integration of dynamic adaptation and contextual security policies, even if they are compromised.

Our experimental evaluation on an NFC-based testbed demonstrated that M-CARBAC successfully prevents unauthorized access attempts in distinguishing legitimate requests from relay attacks. Moreover, while our model incurs a 3% CPU overhead compared to traditional RBAC, this tradeoff is acceptable given the significant security improvements.

Compared to existing security mechanisms, such as distance bounding, RF fingerprinting, and ambient-based detection, M-CARBAC provides a more comprehensive, application-layer solution that detects relay attacks and adapts dynamically to varying access conditions.

Future research should focus on: Optimizing processing time by refining the dual-phase authentication process and exploring machine learning-based anomaly detection to improve attack resistance. These refinements will strengthen M-CARBAC's security, efficiency, and usability, making it a more adaptable and scalable security model for modern NFC-enabled environments.

7 REFERENCES

[1] Anggoro, O., Dzulfikar, M., Purwandari, B., & Mishbah, M. (2019). Secure smartphone-based NFC payment to prevent man-in-the-middle attack. *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. https://doi.org/10.1109/icimcis48181.2019.8985191

[2] Cavdar, D. & Tomur, E. (2015). A practical NFC relay attack on mobile devices using Card Emulation Mode. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. https://doi.org/10.1109/mipro.2015.7160477

[3] Chabbi, S., Boudour, R., & Semchedine, F. (2017). A secure protocol, based on Iris technology, for NFC Phone Applications. *2017 International Conference on Mathematics and Information Technology (ICMIT)*. https://doi.org/10.1109/mathit.2017.8259699

[4] Drimer, S. & Murdoch, S. J. (2007). Keep your enemies close: distance bounding against smartcard relay attacks. *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, *Article 7*, 16.

[5] Dullink, W. van & Westein, P. (2013). *Remote relay attack on RFID access control systems using NFC enabled devices*.

[6] Francis, L., Hancke, G., Mayes, K., & Markantonakis, K. (2010). Practical NFC peer-to-peer relay attack using mobile phones. *Radio Frequency Identification: Security and Privacy Issues*, 35-49. https://doi.org/10.1007/978-3-642-16822-2_4

[7] Francis, L., Hancke, G., Mayes, K., & Markantonakis, K. (2012). Practical relay attack on contactless transactions by using NFC mobile phones. *Radio Frequency Identification System Security: RFIDsec'12 Asia Workshop Proceedings*, *8*, 21-32. https://doi.org/10.3233/978-1-61499-143-4-21

[8] Gurulian, I., Akram, R. N., Markantonakis, K., & Mayes, K. (2017, April 3). *Preventing relay attacks in mobile transactions using infrared light*. https://doi.org/10.1145/3019612.3019794

[9] Gurulian, I., Shepherd, C., Frank, E., Markantonakis, K., Akram, R. N., & Mayes, K. (2017, August 1). *On the Effectiveness of Ambient Sensing for Detecting NFC Relay Attacks*. https://doi.org/10.1109/TRUSTCOM/BIGDATASE/ICESS.2017.218

[10] Isnan, M. I., Putrada, A. G., & Abdurohman, M. (2019). *Detection of Near Field Communication (NFC) Relay Attack Anomalies in Electronic Payment Cases using Markov Chain. 2019*. https://doi.org/10.1109/ICIC47613.2019.8985894

[11] *ISO/IEC 18092:2023. (2023, July 8). ISO/IEC 18092:2023. Telecommunications and Information Exchange Between Systems - Near Field Communication Interface and Protocol 1 (NFCIP-1). https://www.iso.org/standard/82095.html*

[12] Li, P., Fang, H., Liu, X., & Yang, B. (2017). A countermeasure against relay attack in NFC payment. *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, 1-5. https://doi.org/10.1145/3018896.3025144

[13] Kamran, A., Mateen, A., Anwar, M., Raza, B., Ahsan, M., Naeem, W., Asim, Y., & Iqbal, M. (2017). A comparison of collaborative access control models. *International Journal of Advanced Computer Science and Applications*, *8*(3). https://doi.org/10.14569/ijacsa.2017.080340

[14] *NFC Forum*. (2024, July 8). *NFC Forum*. https://nfc-forum.org//

[15] Weiser, M. (1991). *The Computer for the 21st Century*. *265*(3). https://doi.org/10.1038/SCIENTIFICAMERICAN0991-94

[16] Mousa, M. & Dofe, J. (2024). Enhancing NFC/RFID System Security through Accelerometer-Generated Dynamic Keys. *2024 European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. https://doi.org/10.1109/EuCNC/6GSummit60053.2024.10597077

[17] Abubaker, R. & Gong, G. (2024). Relay Attack Detection using OFDM Channel-Fingerprinting. *2024 European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. https://doi.org/10.36227/techrxiv.170630525.51598285/v1

[18] Yang, Y., Xun, Y., Lv, T., & Liu, J. (2024). NFC-RFAE: Semi-supervised RF Authentication for Mobile NFC Card System. *2024 International Wireless Communications and Mobile Computing(IWCMC)*. https://doi.org/10.1109/IWCMC.2024.XXXXXXX

[19] Wang, Y., Zou, J. & Zhang, K. (2023). Deep-Learning-Aided RF Fingerprinting for NFC Relay Attack Detection. *Electronics* 2023, *12*, 559. https://doi.org/10.3390/electronics12030559

Contact information:

Davut CAVDAR, Dr.
(Corresponding author)
Middle East Technical University
Ankara, Turkiye
E-mail: davutcavdar@gmail.com

Emrah TOMUR, Dr.
Nokia,
Munich, Germany
E-mail: emrah.tomur@gmail.com

Aysu Betin CAN, Associate Professor, Dr.
Computer Science - Colorado School of Mines
Colorado, USA
E-mail: aysu.betincan@mines.edu