International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:4, 2017

Design and Application of NFC-Based Identity and Access Management in Cloud Services

Shin-Jer Yang, Kai-Tai Yang

***Abstract*—**In response to a changing world and the fast growth of the Internet, more and more enterprises are replacing web-based services with cloud-based ones. Multi-tenancy technology is becoming more important especially with Software as a Service (SaaS). This in turn leads to a greater focus on the application of Identity and Access Management (IAM). Conventional Near-Field Communication (NFC) based verification relies on a computer browser and a card reader to access an NFC tag. This type of verification does not support mobile device login and user-based access management functions. This study designs an NFC-based third-party cloud identity and access management scheme (NFC-IAM) addressing this shortcoming. Data from simulation tests analyzed with Key Performance Indicators (KPIs) suggest that the NFC-IAM not only takes less time in identity identification but also cuts time by 80% in terms of two-factor authentication and improves verification accuracy to 99.9% or better. In functional performance analyses, NFC-IAM performed better in salability and portability. The NFC-IAM App (Application Software) and back-end system to be developed and deployed in mobile device are to support IAM features and also offers users a more user-friendly experience and stronger security protection. In the future, our NFC-IAM can be employed to different environments including identification for mobile payment systems, permission management for remote equipment monitoring, among other applications.

***Keywords*—**Cloud service, multi-tenancy, NFC, IAM, mobile device.

I. Introduction

DUE to fast developing information and booming data volumes, enterprises are searching for more effective schemes to deal with their large volumes of computing and data. Cloud computing is an attractive choice due to its capacity for processing large amounts of data [1], also cloud computing is a service platform which virtualizes and optimizes IT resources including computing, storage, network, hardware, software, allowing for quantifiable and billable services for on-demand access through the Internet [2].

Multi-tenancy technology is very important in cloud services especially with SaaS. As information security and personal privacy are receiving more attention, cloud multi-tenancy environments will come under stricter scrutiny. Despite its simplicity and widespread adoption, password-based authentication suffers from weaker security relative to two-factor authentication schemes. The problem with the latter is its unfriendly use. While many user accounts have installed two-factor authentication protection they suffer from a poor user experience due to the cumbersome and complex process. Some users may ultimately abandon its use and leave their accounts under more dangerous exposure.

The NFC-based scheme of Lee et al. balances user experience [3] and identity security through the NFC function of mobile devices and an RSA Digital Signature [4]. This method relies on a computer browser program and requires connecting a card reader to a computer for authentication. Its access management is insufficient too, as it lacks good user permission management after successful authentication. This paper designs a new NFC-based Identity and Access Management system, the NFC-IAM, which eliminates reliance upon a computer and a card reader for authentication while offering improved access management. Hence, the main proposes of this paper are as follows:

- Cloud IAM is getting more attention when enterprises flock to adopt cloud services. Yet a balanced user experience and secure IAM scheme on mobile devices remains rare.
- We developed an NFC-IAM system called NFC-IAM for cloud third-party.
- Front-end app and back-end systems based on NFC-IAM is an IAM system on mobile devices with balanced security protection and good user experience.
- The authentication time and accuracy KPIs are compared and analyzed against other schemes with simulation experiments.
- Security of the NFC-IAM based system is analyzed with a threat model.
- We defined and compared the functional performance including: scalability, portability, efficiency, physical -effortless, recoverability, and user cost against another scheme.

This paper is organized as follows. In Section I, this paper introduces the research backgrounds and purposes. Section II will survey and describe the review of related works and current issues in multi-tenancy and its IAM. Section III will examine operations of NFC-IAM scheme and design its algorithm. Section IV comprises the set of simulation experiments and analyzes the results in KPIs and functional performance. A conclusion will be drawn in Section V.

II. Related Work

A. Multi-Tenancy and Its IAM

In cloud multi-tenancy technology, a tenant is a user of a system or computing resources. Cloud service providers rent users (tenants) with application program developed or computing resources to multiple users for access at the same

Shin-Jer Yang, and Kai-Tai Yang are with the Department of Computer Science and Information Management, Soochow University, Taipei 100, Taiwan (e-mail: sjyang@csim.scu.edu.tw, 02356004@scu.edu.tw).

time [5], [6]. This requires a specially designed application program and computing environment.

IAM is an important application in cloud multi-tenancy technology, as application program and computing resources are shared by tenants. This mandates cloud service providers to maintain good application context independence and data isolation [7] among individual tenants and to prevent user benefits from being damaged due to interference. This in turn mandates the use of IAM to determine and validate user identity before availing them with required access permission in the following three steps, including: (1) Validate user identity: The system determines the user's identity with their user ID and password. This is an important way to show proof for granting the user required permission; (2) Grant users required permission: After the system validates user identity, the system further grants users the required permission with user information or attributes. This permission is an important way to prove that users request the services or data in a multi-tenancy environment; and (3) Keep logs of authentication and granted permissions: The logs not only let users check their login and granted permission record, but also allows the system or administrator to inspect it in the future [8].

This IAM is known as Identity as a Service (IdaaS) or Identity Management as a Service in a cloud environment. One of these applications is Single Sign-on (SSO), which enables users to log in and get a certificate with a single identity to access resources and service from different sources by a certificate transmitted with Security Assertion Markup Language (SAML).

B. Two-Factor Identity Authentication Method and Its Application

Two-Factor Authentication (2FA) or Two Step Verification [9] is a technology patented in 1984 which authenticates users with two elements or objects. They can be: something you know, including password or PIN (Personal Identification Number); something you have, including Smart Card, USB Key or PKI (Public Key Infrastructure) certificate; something you are including, fingerprints, iris, retina, face or voice.

Some commonly adopted two-factor identity authentication schemes are:

- SecurID [10]: A user device is given a seed to generate a six-digit authentication code every 30 or 60 seconds continuously. The system server keeps the same seed for each user for authentication. Users who want to log in must provide the seed generated six-digit authentication code and the server authenticates the input against the same seed and synced time.
- 2-steps [11]: The first step of this scheme works the same as a conventional password authentication scheme where the server authenticates the user with an ID and password provided. The next step is usually undertaken by the server sending a one-time authentication code to the user by SMS or push message. The user is required to input it in a login page or program within a set time to complete the authentication process.

C. NFC Overview and NFC-Based Identity Authentication Scheme

NFC is a short distance, high-frequency wireless communication technology which enables contactless peer to peer (P2P) data transmission between electronic devices up to 20cm apart in a frequency of 13.56MHz [12]. Originally from non-contact Radio Frequency Identification (RFID) NFC, it now has multiple global standards including ISO/IEC IS 18092, EMCA-340, and ETSI TS 102 190.

There are two types of NFC equipment [13], including: (1) Active NFC equipment, which must be powered before acting as an initiator to send a connection request to a target NFC equipment by generating a radio frequency field; and (2) Passive NFC equipment, which can act as a target equipment to receive connection requests. Also, NFC equipment consists of three modes with the following features [14]:

- Card Emulation Mode: This is nothing but an IC card with RFID technology.
- P2P Mode: Connect two NFC devices for P2P data transmission in an NFC Data Exchange Format (NDEF).
- Reader/Writer Mode: NFC equipment is used as a contactless reader in this mode.

The NFC-based identity authentication by Lee et al. employs a mobile device's NFC chip for 2FA. Mobile devices generate and save one private key [15] and server information in a mobile device database, while the public key and Universally Unique Identifier (UUID) of the latter is saved after successful registration to the server. To log in to a user account, the user sends a login request to the server from their computer browser. The latter then replies to the user with a public key encrypted message containing data including server information and a timestamp. The computer browser then drives the computer connected card reader and sends the data received to the mobile device. The latter then matches the received server information before digitally signing off the message with the user's private key and returns it to the server along with the UUID through the card reader connected to computer. The server then receives a public key based on the received UUID to decrypt the digitally signed messages to end the login operation.

Despite its stronger security and user experience, the NFC-based 2FA has the disadvantage of requiring a computer and a card reader for authentication. This prevents it from being adopted for identity authentication through mobile devices, not to mention its poor access management capabilities. Hence, this paper designs an NFC-IAM scheme addressing this shortcoming.

III. Operations and Design Issues of NFC-IAM

NFC-IAM is an NFC-based third-party cloud IAM scheme. A cloud application and back-end identity authentication system was developed to carry out IAM with a mobile device in a cloud multi-tenancy environment. It features the following:

- NFC-IAM is a third-party IAM scheme which relies on an Identity Provider (IdP) in managing user identity, granting permission, and providing information to a Service Provider (SP) during user login.

- Users can log in to their account from a mobile device without any need for a computer browser or a card reader.
- After successful login, the NFC-IAM scheme grants users different permissions for access to different resources according to their identity status.
- User IDs and passwords are no longer needed. Thus, the NFC-IAM input process offers greater user convenience and a better user experience.

A. Operation Structure and Description

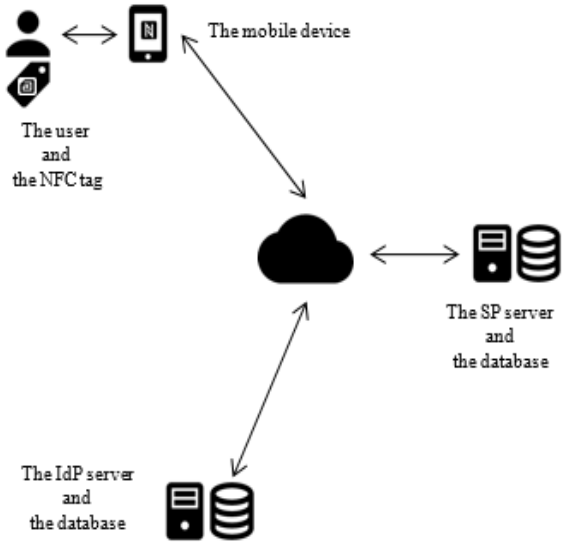

Fig. 1 NFC-IAM operations structure

This study builds up a third-party cloud IAM based on NFC-IAM scheme. The client end is composed of a user's mobile device and an NFC tag while the server end consists of a third-party IdP server for IAM and SP server for resources and service access, as shown in Fig. 1. The NFC-IAM operational process runs in three phases: (1) Users register with the IdP server; (2) Users register with the SP server through the IdP server; and (3) Users log in to the SP server through the IdP server as shown in Fig. 2. The process flow details are described below.

1) Phase 1: Users Register with the IdP Server

Based on the operational flow as shown in Fig. 3, the detailed operations of phase 1 are illustrated as follows:

a) When registering with the IdP server, the user's mobile device will acquire a pair of public and private key *PbK* and *PvK* through an RSA encryption algorithm.
b) The user's mobile device sends the user's registration request to the IdP server and waits for a reply.
c) The IdP server replies with a piece of the information *Info* to the user's mobile device.
d) The user's mobile device displays the received information *Info* on the screen for user validation. The user agrees to proceed with the registration or the process stops here.
e) After the user agreement, the user's mobile device requests to scan a blank NFC tag. If the blank NFC tag is scanned successfully the mobile device writes a *PvK* in NDEF format to the blank NFC tag and saves *Info* replied by the IdP Server to the mobile device database.
f) The mobile device sends the *UUID* and *PbK* to the IdP server.
g) IdP server saves the *UUID* and *PbK* to the database to end the registration process.

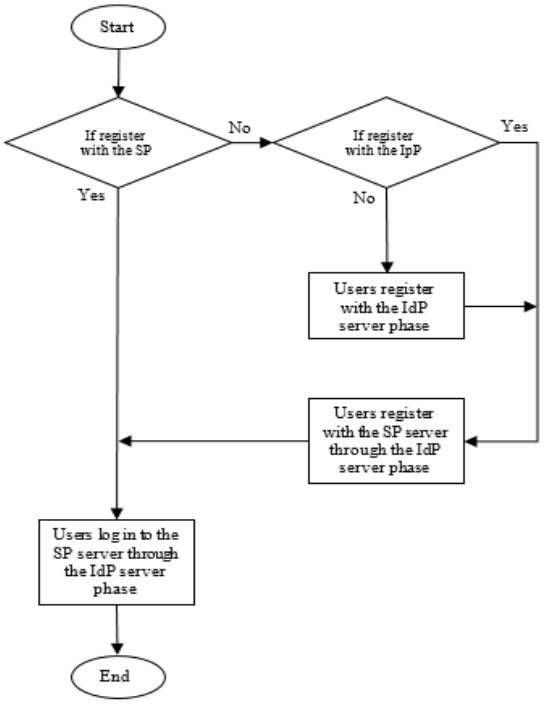

Fig. 2 The operational flow of NFC-IAM

2) Phase 2: Users Register with the SP Server through the IdP Server

Based on the operational flow as shown in Fig. 4, the detailed operations of phase 2 are illustrated as follows:

a) The mobile device sends a request to the SP server and waits for its reply.
b) The SP server replies to the mobile device with the IdP server directories available for registry, and the mobile device displays them to the user for selection.
c) The IdP server of NFC-IAM is selected by the user.
d) The mobile device sends the *UUID* to the IdP server.
e) The IdP server verifies the received *UUID* with the one saved in the IdP server database. If they match, it sends processing results *R* to the SP server and waits for the reply.
f) The SP server queries the user information from the IdP server once it has received processing results *R* and returns the newly received *R* for authentication.
g) The IdP server returns the user information *A* to the SP server once the request has been received and *R* gets authenticated.
h) The SP server receives and saves the user information *A* and replies with the results to the mobile device.

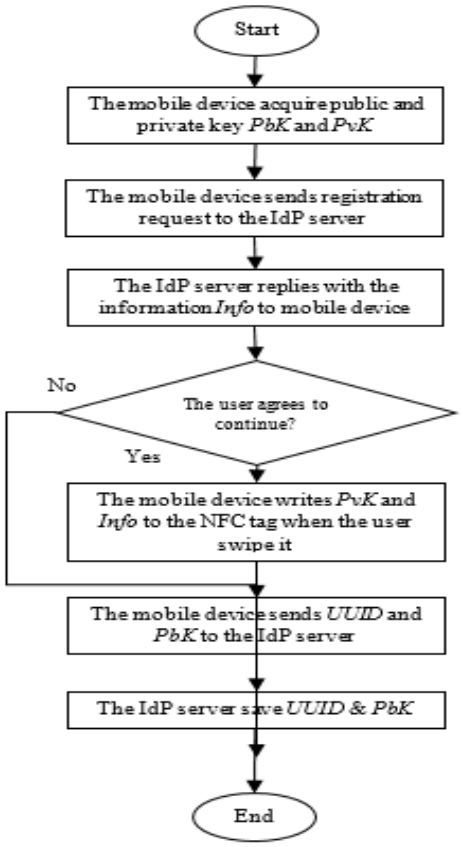

Fig. 3 Operational flow in user registration with the IdP server

3) Phase 3: Users Log In to the SP Server through the IdP Server

Based on the operational flow as shown in Fig. 5, the detailed operations of phase 3 are illustrated as follows:

a) The mobile device sends the login request to the SP server and waits for its reply.
b) The SP server replies to the mobile device with the IdP server directories which are available for registry, and the mobile device displays them to the user for their selection.
c) The IdP server of the NFC-IAM is selected by user.
d) The mobile device sends a *UUID* to the IdP server.
e) The IdP server verifies the received *UUID* against the one saved in the IdP server database. If they match, the IdP server returns information *Info*, timestamp *T* and random number *N* to the mobile device, to complete the login process; or it returns a login failure message and ends the login process.
f) Once the information *Info*, timestamp *T* and random number *N* have been received, the user must check whether the *Info* received matches with the one saved in the mobile device database. The login process proceeds if *Info* matches, otherwise, it ends.
g) The mobile device prompts the user to scan an NFC tag. The user must scan the one used for registering to retrieve the private key *PvK* for the RSA digital signature to get *EncN* from the received random number *N*.
h) The mobile device returns timestamp *T* and *EncN* back to the IdP server.
i) The IdP server uses the *UUID* to get the user's public key *PbK* saved in the database, uses *PbK* to decrypt *EncN* into *DecN*. The IdP server matches timestamp *T* and original *N* and *DecN*. The login process proceeds if both data match; otherwise it ends.
j) The IdP server replies with the processing results *Q* to the SP server and requests user information, and returns the newly received *Q* for authentication.
k) After the SP server request has been received and *Q* has been authenticated, the IdP server returns user information *B* to the SP server.
l) The SP server matches user information *B* to user information *A* in the database before granting the user login permission *P*.
m) The SP server returns login results to the user and ends the login process.

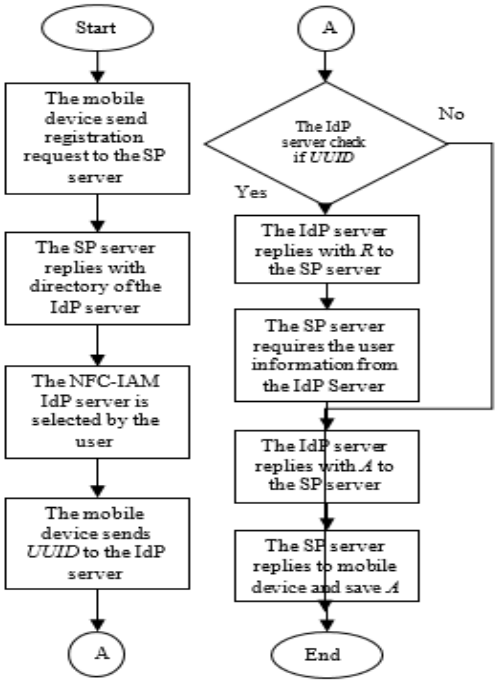

Fig. 4 Operational flow in user registration with the SP server through the IdP server

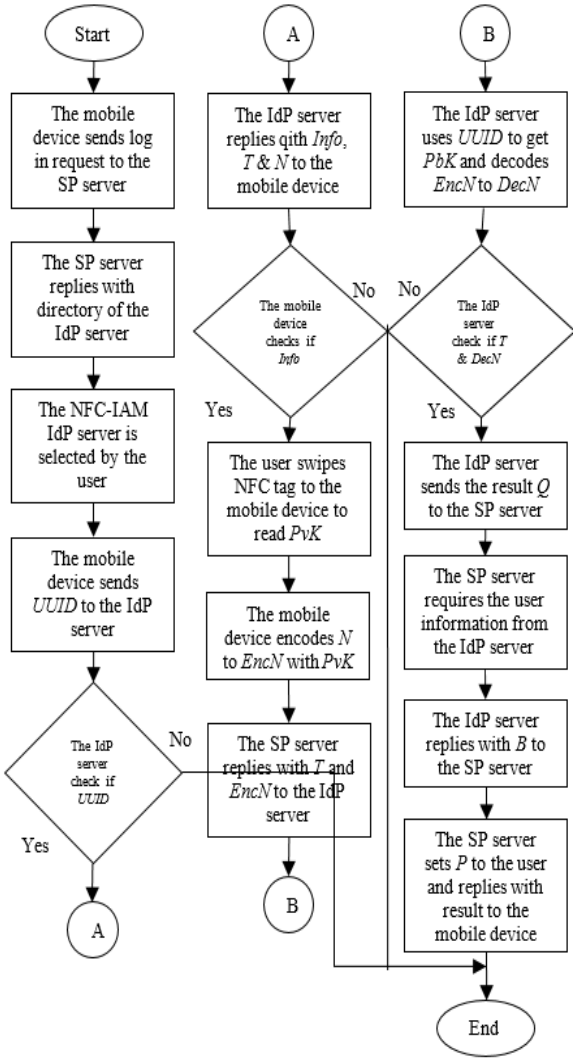

Fig. 5 Operational flow in users log in to the SP server through IdP the server

B. Algorithm Design

Algorithm pseudo code according to Fig. 2 and given three layers is described below:

Algorithm: NFC-IAM

Input:
#define *PbK*: Public key
#define *PvK*: Private key
#define *info*: The IdP server's information
#define *A*: User's information from the IdP server
#define *B*: User's information from the IdP server
#define *R*: Result of process
#define *Q*: Result of process
#define *P*: Permission
#define *T*: Timestamp
#define *N*: Random number
#define *EncN*: Encrypted random number
#define *DecN*: Decrypted random number
#define *UUID*: The UUID of the mobile device
#define register_with_IdP(): Register with IdP server
#define register_with_SP(): Register with SP server
#define log_in_to_IdP(): Log in to IdP server
Output: To complete NFC-IAM Method.
Method:

```
1:  Begin
2:    if The user registers with the SP server then
3:      call Procedure log_in_to_SP()
4:    else
5:      if The user registers with the IdP server then
6:        call Procedure register_with_SP()
7:      else call Procedure register_with_IdP()
8:      end if
9:    end if
10: End
11: Procedure: register_with_IdP()
12:   The mobile device acquires PvK and PbK
13:   The mobile device sends registration request to IdP server
14:   The IdP server return Info
15:   if The user checks Info and agrees to continue then
16:     The mobile device writes PvK and Info to the NFC tag
17:     The mobile device sends UUID and PbK to IdP server
18:     The IdP server saves UUID and PbK in to the database
19:   end if
20: End procedure register_with_IdP()
21: Procedure: register_with_SP()
22:   The mobile device sends registration request to SP server
23:   The SP server return directory of the IdP server
24:   if The user selected the NFC-IAM IdP server then
25:     The mobile device sends UUID to the IdP server
26:     if The IdP server checks UUID exist then
27:       The IdP server sends R to the SP server
28:       The SP server requires for user's information
29:       The IdP server return A to the SP server
30:       The SP server saves A in to the database
31:       return result to the mobile device
32:     end if
33:   end if
34: End procedure register_with_SP()
35: Procedure: log_in_to_SP()
36: The mobile device sends log in request to the SP server
37:   The SP server return directory of the IdP server
38:   if User selected the NFC-IAM IdP server then
39:     The mobile device sends UUID to the IdP server
40: if The IdP server checks UUID exist then
41:       The IdP server return Info, N and T
42:       if The mobile device checks Info is matched then
43:         The user swipes the NFC tag to read PvK
44:         The mobile device encodes N to EncN with PvK
45:         return T and EncN to the IdP server
46:         The IdP server uses UUID to get PbK to decodes EncN
47:         if N equals to DecN then
48:           The IdP server sends result Q to the SP server
49:           SP server requires user's information
50:           The IdP server return B to the SP server
51:           The SP server sets P to user
52:           return result to the mobile device
53:         end if
54:       end if
55:     end if
56:   end if
57: End procedure log_in_to_SP()
```

END NFC-IAM.

IV. Simulation Environments and Result Analysis

A simulated system based on this NFC-IAM scheme will be analyzed in terms of KPIs of authentication time and accuracy and security first. It is then compared against popular identity authentication methods of the mobile device with respect to function and benefit.

A. Experiment Environment Configuration

The experimental environment is built to simulate, test, and analyze the identity authentication application and back-end system we built upon the NFC-IAM scheme. Hence, Tables I-III present for specification details of the simulated environment.

TABLE I
NFC Tag Simulation Environment

Software / Hardware	Specification
Standard	ISO/IEC14443A
Type	Type 4
Size	4KB

TABLE II
Mobile Device Simulation Environment

Software / Hardware	Specification
Model Number	HTC Desire 816
Operation System	Android 5.0.2
CPU	4Cores
Memory	1.5GB
Disk	8GB
NFC	ISO- 14443- 2

TABLE III
IdP Server and SP Server Simulation Environment

Software / Hardware	Specification
Operation System	Ubuntu 15.04
CPU	2 Cores
Memory	4GB
Disk	500GB
MySQL Version	MySQL 5.7.6
PHP Version	5.6.0
Apache Version	2.2.31
JDK Version	1.8.0-65

B. Authentication Time and Accuracy KPI Analysis

The security analysis of the third-party NFC-IAM system has been covered in the last section. Here we review the authentication time and accuracy with details shown in Table IV.

The schemes of pure password (PWD), two-step authentication code entry (2STP), and NFC-IAM are compared with each other in terms of authentication time. PWD by nature outperforms 2STP and NFC-IAM as the latter two are 2FA based. The goal is to show that NFC-IAM offers better security protection with relatively acceptable longer authentication time than PWD does.

We simulate the execution with Java code's system class method on 5000, 10000, and 20000 times. Execution times are averaged in milliseconds (*ms*) with results as shown in Table V. It is clear NFC-IAM is outperformed by PWD by around 25%, which is acceptable. However, NFC-IAM betters 2STP by some 80% as shown in Fig. 6.

TABLE IV
Key Performance Indicators

KPIs	Purposes
Authentication time	Calculated the average time from receipting the request of log in to returning user's information and permission.
Accuracy	Calculated the accuracy of IAM when users log in.

TABLE V
Experimental Results

KPIs	Times Scheme	5000	10000	20000
Authentication time	PWD	740.85	793.44	766.59
	NFC-IAM	985.66	997.89	1042.91
	2STP	1934.78	1969.42	1902.88
Accuracy	PWD	99.96	99.96	99.95
	NFC-IAM	99.98	99.98	99.97
	2STP	99.98	98.97	99.97

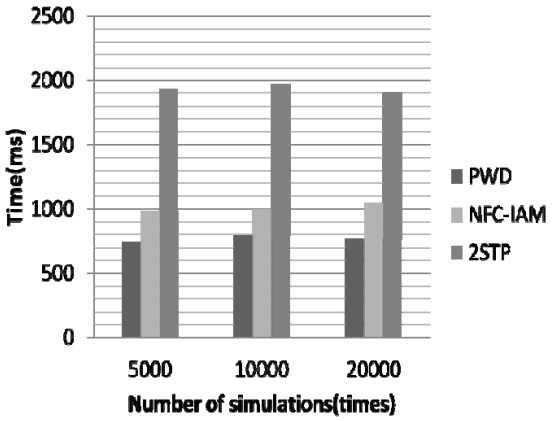

Fig. 6 Authentication time comparisons

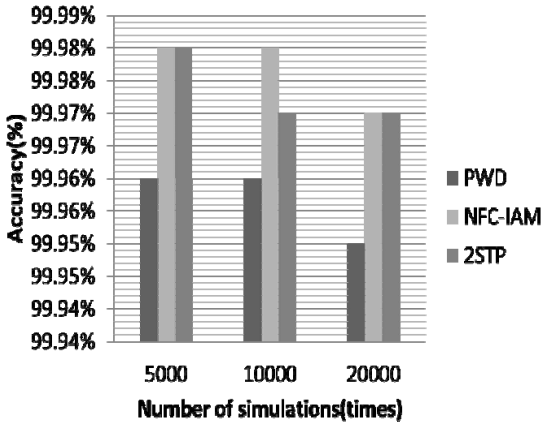

Fig. 7 Accuracy comparisons

The same environment is adopted for comparing the authentication accuracy of these three schemes. The rates of the correct authentication in 5,000, 10,000, and 20,000 executions are shown in Table V. Though no sharp differences exist among the three of them, NFC-IAM enjoys an accuracy rate of over

99.9% as shown in Fig. 7.

C. Security Analysis

A threat model of Asset, Trust Boundaries, and Analysis [16] Threat is adopted to analyze the IAM cloud application and back-end system based on NFC-IAM.

1) Assets

The NFC-IAM based cloud application and back-end is an IAM system aimed at protecting user privacy and login authenticity. This requires safeguarding the user's private key and preventing authenticating random number N before and after signoff from retrieval by an attacker at the same time. Assets to be protected by the IAM system are set to:

- User authentication information, i.e., the private key saved in the NFC card, which mandates good protection.
- Authentication information in IdP and SP servers are being secured to prevent an attackers' retrieval.
- Permission data acquired by the user after successful identity authentication requires protection too.

2) Trust Boundaries

The NFC-IAM based system is composed of six components along with data transmission channels as shown in Fig. 8. Our assumptions and analysis are:

- NFC tag: This is usually carried along by users. Data in it protected with a one-time only write-in, private key is useless without the UUID in the user's mobile device.
- Mobile device: The NFC-IAM application runs on the user's mobile device without sharing with others. We can assume that the database and memory in the user's mobile device has no chance to be tampered with under normal use.
- Identity and resource database: Data in it cannot be accessed directly from the external environment. The IdP server uses the UUID to match the user data in its identity database, while the SP server acquires the database permission through the user information in the IdP server.
- Communication between the NFC tag and the mobile device: This is not a safe channel and may be susceptible to data breaches in between.
- Network channel within each component: Network channels are all Transport Layer Security (TLS) encrypted.

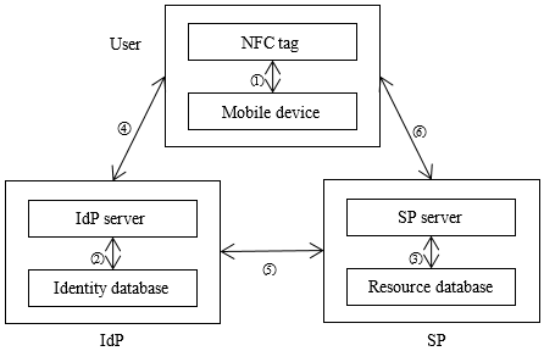

Fig. 8 NFC-IAM components structure

3) Threat Analysis

Threats existing in system components and communication processes [17] are itemized and analyzed below:

- Loss of mobile device: Lost mobile device may expose the UUID contained in it which is not a threat as the private key cannot be acquired the same way the public key may be acquired from the IdP server with the UUID. This secures the data even in the event of loss of the mobile device.
- Mobile device screen being watched: The NFC-IAM scheme does not require ID and password entry for login. This eliminates the possibility of ID theft by peeping into the mobile device screen.
- NFC tag may get copied or tampered: NFC tag can be set to a one-time write-in. In case it is copied, the user's private key may be exposed. Yet the public key aligned with server can be acquired only with the UUID in the user's mobile device.
- NFC communication vulnerability: The mobile device and NFC tag used by the user for login are both possessed by the same person and should be free from information leak due to login with different devices. In addition, NFC acts within too short of a time span to leave attackers any chance for information capturing. The short communication distance of NFC makes data capturing much more difficult.
- Reverse-compile the app to get connection information: Users make requests to the SP server for resources or data access while the latter replies redirecting the users' logging into the IdP server. This leaves no IdP server connection data in the application's source code. There is no chance to get the connection data by reverse-compiling. The private key is randomly generated by the mobile device rather than hard coded in the application. Reverse compiling the application does not reveal the private key to attackers.
- Attacks on the SP server and the IdP server database: There is user data and permission access control contained in the database. The login to the SP server and acquiring identity permission and access management is authenticated by a third-party IdP server. The user's public key and login information can be accessed on the IdP server only through the relevant UUID.
- Middleman attack: Each network connection is TSL encrypted to prevent tapping by a middleman attack. Identity authentication information is secured from tapping as long as the encrypted certificate is kept in mobile device safely.
- Phishing: The mobile device will validate the IdP server information, display relevant information to the user, and get the user's consent before logging in. This protects users against phishing and prevents information from being exposed.

D. Functional Performance Analysis

We compared and analyzed the functional performance of NFC-IAM with PWD, SecurID, and 2STP. The performance indexes include: scalability, portability, efficiency, correctness,

recoverability and user cost. The details are shown below:

- Scalability: This criterion measures the scalability of the authentication schemes in terms of the burden placed on the users by each service.
- Portability: An authentication scheme would be considered high portability if it does not require the user to bring anything except the mobile device.
- Efficiency: The efficiency is represented by the execution time during an authentication process, including the calculation time and the operation time.
- Physical-effortless: This criterion measures how much physical effort a user should make in an authentication process.
- Recoverability: This criterion checks the recoverability of the authentication scheme if a user forgets the required credentials or loses the device.
- User cost: This criterion measures the total cost per user of the scheme.

In the last paragraph, we defined several performance indexes, and the following is our analysis:

NFC-IAM is a third-party identity access management scheme, and its back-end is an IdP. In other words, once users register with the NFC-IAM IdP server, users can register or log in and get resources from different the SP servers by passing the identity information with the NFC-IAM IdP. Therefore, the NFC-IAM scheme has high scalability. Users can authenticate their identity anywhere and anytime by only using an NFC enabled mobile device, and an NFC tag. The volumes of both things are very convenient for users to carry. They are not too great a burden to users, and thus, the NFC-IAM scheme has high portability.

It is easier for the user to log in to their account by replacing the user ID and password or an authentication code. Users only have to scan an NFC tag when they login. Therefore, it is more efficient to authenticate through the NFC-IAM scheme. For the proposed scheme, users only need to swipe their mobile device at the NFC enabled mobile device, but for 2STP or SecurID, users authenticate with more than one physical effort.

TABLE VI
FUNCTIONAL PERFORMANCE COMPARISONS

	PWD	NFC-based	NFC-IAM	SecurID	2STP
Scalability	X	X	O	X	O
Portability	O	X	O	O	O
Efficiency	O	O	O	X	X
Physical - effortless	X	O	O	X	X
Recoverability	O	X	X	O	O
User cost	O	X	X	X	X

It is obvious that users cannot easily recover the account if their mobile device or NFC tag is broken or lost. Actually, all token-based and device-based schemes have poor performance in the recoverability criterion. The NFC-IAM scheme needs an NFC enabled mobile device; therefore, it will increase the user cost at the present moment. But we assume that this kind of mobile device will become more popular, and the user cost of the NFC-IAM scheme will be lower.

To combine the above descriptions, we have to iterate through Table VI. ("O" or "X" represents "Good" or "Poor")

V. CONCLUSION

In this paper, we designed an NFC-IAM scheme to develop an NFC-IAM based front-end application and back-end IAM system to IAM in a multi-tenancy cloud environment. Also, we analyzed the security level of NFC-IAM, and compared it with other types of identity authentication schemes. The threat model analysis with attack assumptions demonstrates that the NFC-IAM based system can defend against most information security attacks. In addition, relative to other identity authentications, the simulation results indicate that NFC-IAM outperforms other 2FA schemes by 80% in terms of efficiency (less time), while maintaining an accuracy above 99.9%.

The NFC-IAM scheme executes identity authentication without a computer and browser, while offering an additional access management function capable of granting differentiated access to different resources based on user permissions. The NFC tag can be read without the need for an external card reader. Although the NFC-IAM scheme has increased user cost since it needs an NFC-enabled mobile device, we assume that this kind of mobile device will become more popular, and the user cost of the NFC-IAM scheme will be lower in the near future. Future directions for applications of this identity authentication scheme lies in different fields, for instance, the emerging field of Fin-Tech, which requires combined IAM for mobile and third-party payments, other fields being the fast growing Internet of Things, and reliable IAM functions for assisting remote monitoring systems and devices.

ACKNOWLEDGMENTS

The partial work of this paper is funded and supervised by the Ministry of Science and Technology in Taiwan under Grant MOST 105-2410-H-031 -032 -.

REFERENCES

[1] National Institute of Standards and Technology, "The NIST Definition of Cloud Computing," In *National Institute of Standards and Technology*, 2011.
[2] Xiao-Yong Li, Yong Shi, Yu Guo, Wei Ma, "Multi-Tenancy Based Access Control in Cloud," In *Proceedings of 2010 International Conference on Computational Intelligence and Software Engineering* (*CiSE*), 2010.
[3] Lee, Haw; Hong, Wei-Chih; Kao, Chia-Hung; Cheng, Chen-Mou, "A User-friendly Authentication Solution using NFC," In *Proceedings of 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, 2014.
[4] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" *Communications of the ACM 21*, pp 120-126, 1978
[5] Krebs, Rouven, "Architectural Concerns in Multi-tenant SaaS Applications," In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science*, 2012.
[6] "Multi-Tenant Data Architecture," https://msdn.microsoft.com/en-us /library/aa479086.aspx, June 2006.
[7] D. Linthicum, "The silly debate over multitenancy," http://www. infoworld.com/article/2683529/clowasud-computing/, 9 Apr 2010.
[8] R. Morris and K. Thompson, "Password security: a case history," vol. 22, pp. 594-597, Nov. 1979.

[9] B. Chess and B. Arkin, "The Case for Mobile Two-Factor Authentication. Security & Privacy", *IEEE* vol. 9, no. 5, pp 81-85, 2011
[10] "RSA SecurID," http://www.emc.com/security/rsa-securid.htm.
[11] "Google 2-step verification," http://www.google.com/landing/2step/, 2013.
[12] D. Rinner, H. Witschnig, E. Merlin, "Broadband NFC - A System Analysis for the Uplink," *Information Forensics and Security*, pp 292-296, 2009.
[13] Zheng-Qin Jian, Yu-Chung Huang, Jehn-Ruey Jiang, "A Privacy Preserving NFC Guestbook System," http://staff.csie.ncu.edu.tw /jrjiang/publication/CSIT2015(NFC-Guestbook).pdf.
[14] E. Haselsteiner and K. Breitfuß, "Security in Near Field Communication (NFC): Strengths and Weaknesses," *Workshop on RFID security*, pp 1-10, 2006.
[15] C. Lu, A. L. M. Santos, F. R. Pimentel, "Implementation of Fast RSA Key Generation on Smart Cards" In *Proceedings of the 2002 ACM Symposium on Applied computing*, pp 214-220, 2002.
[16] Web application threat model. http://msdn.microsoft.com/en-us/library/ ms978531.aspx.
[17] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: a framework for comparative evaluation of Web authentication schemes," University of Cambridge, Computer Laboratory, Tech. Rep. 817, March 2012.