Nagaraju and Parthiban *Journal of Cloud Computing: Advances, Systems and Applications* (2015) 4:22
DOI 10.1186/s13677-015-0046-4

Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH **Open Access**

Trusted framework for online banking in public cloud using multi-factor authentication and privacy protection gateway

Sabout Nagaraju* and Latha Parthiban

Abstract

Cloud computing is an emerging, revenue generating and internet based technology, where computing power, storage and other resources are provided to the stakeholders in a single package. The traditional online banking systems can make use of the cloud framework for providing economical and high-speed online service to the consumers. This paper first describes a systematic Multi-factor bio-metric Fingerprint Authentication (MFA) approach which provides a high-secure identity verification process for validating the legitimacy of the remote users. The significance of this approach is that the authentication credentials of the users are not revealed to the bank and cloud authentication servers, but allows the servers to perform remote users' authentication. We then extend this investigated framework to develop a privacy protection gateway for obscuring and desensitizing the customers' account details using tokenization and data anonymization techniques. This approach retains the original format of data fields at various levels of the database management systems and makes the data worthless to others except the owner. In addition to designing an efficient MFA, through extensive experimental results we illustrate our privacy protection gateway is practical and effective.

Keywords: Public cloud, Online banking, Fingerprint biometrics, Tokenization, Data security and privacy

Introduction

Banks provide the impetus for people and country to develop economically. They make financial dealing easy, safe and convenient. Banks take part in welfare activities and also help in social causes of the people. Most of the banks provide the financial dealings through passbooks, ATM, mobile banking, electronic banking and telephone banking. Among these financial dealings, e-banking and mobile banking will be more convenient and these two are essential for busy people. Specifically, it is critical to provide an efficient, reliable and secure e-banking service to the consumers because user needs and cyber-attacks are increasing on the internet-based technologies. The cloud environment is more suitable paradigm to new, small and medium scale banking organizations as it eliminates the requirement [23] for them to start with small resources and increase gradually as the service demand rises.

One of the most advanced technology today is cloud computing, which provides expert's solutions, computing and storage resources as outsourcing on the pay-per-use basis at nominal cost. In India, Pondicherry Co-operative Urban Bank Ltd, Nawanagar Cooperative Bank Ltd, which is in Gujarat and Sree Charan Souharda Cooperative Bank Ltd, located in Bangalore are currently using IBM Smart Cloud and HDFC retail banking moved to Oracle Private Cloud (i.e., Oracle Enterprise Manager 12c). Bank of America and Merrill lynch also moved to IBM cloud banking services. Microsoft organisation has spent billions of dollars for providing cloud services and they have tie-up with TEMENOS organisation to offer core banking services to the stakeholders [25]. TATA Consultancy Services (TCS) reported that cloud computing is the next generation banking technology with innovative business models [26].

* Correspondence: nagarajus.ucc@pondiuni.edu.in
Department of Computer Science, Pondicherry University Community College, Lawspet, Pondicherry 605 008, India

The transformation of the banks into cloud computing has following advantages:

- Can provide optimised, virtualized and scalable operational environments.
- Banks can overcome present and future challenges.
- High speed bandwidth can be provided to access online banking services in milliseconds.
- Banking services can cover geographically with effective multi-channels integration.
- Banks can be more attractive for providing new offerings.
- Revenue gain for new, small and medium scale banking.
- High cost of running in-house data centres can be eliminated.
- Can provide flexible platforms to build and bring advanced banking services to public.

The problems and risks

Today banking organizations are operating with high competitions, bank brand names, and regulated environments. Therefore, the aspects of core banking services are influenced by business considerations and compliance requirements. Innovations done so far in bank technologies, operations and security controls have been managed inside the enterprises. Since, the public cloud based functionalities and security control aspects are managed out of the banking enterprises. These out of boundary aspects have highly influenced on the bank adoption and sometimes this adoption may damage brand name and existence. Data security and privacy concerns prevent the banking stakeholders to migrate to the cloud-based platforms. As part of the threats landscape within public cloud, the online banking services need to be protected from the cloud service provider and other malicious attackers.

The following are the biggest and legitimate problems associated with public cloud solutions:

1) Making sure that user access keys and credentials are secured. Access credentials and keys for the cloud-based infrastructure that rented out from some public cloud vendors need to be appropriately managed and protected. Four major attacks envisioned in our proposed authentication process, they are:
    i. Malicious insiders of the cloud and bank cannot learn the remote user credential parameters.
    ii. User login and authentication credentials are not revealed to the cloud and bank servers.
    iii. An attacker may eavesdrop on the credential communication channel and he/she may use replay attack.
    iv. Sometimes, an attacker may change the network IP of the authorized user so that the request coming from that altered system appears to be a request coming from an impersonated user.
2) Similarly, dependency on geographic or legal jurisdiction that becomes another added point to consider, because certain laws in certain political jurisdictions may allow certain local agencies to access to the data that is hosted within their territory. For instance, the patriot law in the United States allows certain US agencies to demand access to the data which is stored in the US Union Territory. Banking information systems are sensitive to this kind of situation. So, there is a need to take appropriate measures to make sure that banking information still remains private regardless of whether it is hosted in any territory or not.
3) In the similar manner, multi-tenancy, where multiple consumers are hosted in a shared public cloud infrastructure for instance. There are chances that they may interfere with each other by some manner.

From the banking organizations perspective several risks are associated with public cloud solutions. Some of the key risks are summarized below.

i. Complexity in banking governance, compliance and audit management
ii. Dilution in bank functional, operational and technology control aspects can lead impact on reputation, regulatory and business.
iii. Difficulties in sustaining security standards, regional privacy laws and information acts.
iv. Banking services will be locked in cloud and it is difficult to bring back in-house if required.
v. Potentially cloud API's are lacking in portability, so stakeholders cannot move from one cloud service provider to another.

Our contribution

The following are the major contribution of our research.

1) *Multi-factor Authentication*: The multi-factors like user ID and password, biometric fingerprint and random strings are used as key parameters in authentication process. Here, user ID and password show what user knows, fingerprint biometric represents who the user is, and random strings are used for verifying the user's identity to servers, server's identity to the user and servers identity to other servers. Our proposed MFA provides a convenient and high-secure multi-stage identity verification process using random strings.

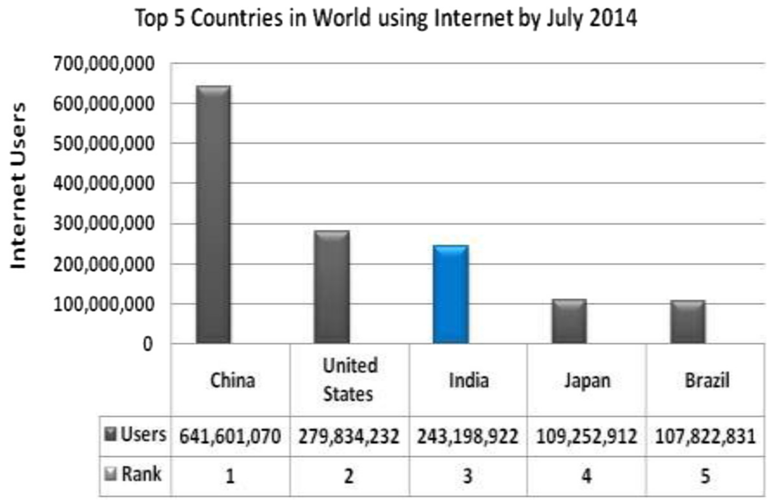
2) *Strong privacy preservation of user credentials*: In our proposed authentication scheme, user credentials are not stored in cloud servers but allow the servers to perform authentication on hashed credentials. Moreover, the cloud service providers itself is corrupted still cannot learn the user credentials.
3) *Authorization protocol*: We propose an authorization protocol which provides data access tokens for each authenticated user to access the account details from the public cloud servers.
4) *Strong Data Privacy*: Data protection issues like data privacy, residency and compliance laws are achieved by using various tokenization techniques and privacy preservation mechanisms.
5) *Provable Security*: Our proposed multi-factor biometric fingerprint authentication and protection gateway mechanisms provide true protection for the user credentials and sensitive account details in a public cloud. Therefore the problems and risks associated with public cloud can be eliminated.

This paper is further divided into seven sections. Introduction provides the background information required to understand the present and future problems associated with in-house online banking systems. The overview of our trusted framework is summarized in Motivation. Our trusted framework describes the authentication, authorization and privacy protection mechanisms. Completeness of our proposed authentication protocol analysis is described in Our protection mechanisms. Completeness of authentication protocol reports the experimental study of our proposed schemes. Literature reviews associated with our research work are presented in Experimental evaluation. Related work summarizes the proposed methodologies and future directions.

Motivation

In this section we provide the essential information required to understand present and future problems associated with in-house online banking systems. Figure 1 represents the statistical information of top 5 countries internet usage by July 2014. Because of the arrival of new mobile internet technologies and other broad-band internet technologies [30] internet users are increasing gradually. The projected global internet users of top five countries by 2011 and 2015 are reported in Fig. 2. With this growth, we can say that internet is becoming daily utility to all groups of people and most of them are using online banking services for fund transfer, e-bills payment, e-shopping and other facilities listed in [5, 20]. In recent years, efforts are being made to develop e-banking for e-commerce sectors. In [1, 2] authors are reporting that e-banking users are growing exponentially in recent years. In china, e-banking usage has increased in trillion yuan from 2008 to 2015. Globally 47 % of customers are willing to use e-banking for their daily needs [23]. Globally 1 in 4 internet users access online banking sites [34]. The e-banking has various advantages for both consumer and banks as given in Table 1. In [20], author Jagdeep Singh describe various e-banking advantages to the government, nation, merchants and others.

Customer's point of view high level security, advanced features and user friendly technologies are the main considerations. Providing these requirements is highly expensive. Many IT projects have been failed in the banking sectors due to lack of understanding user's requirements and technology illiteracy. Hence there is a tremendous demand for outsourcing core banking in many



	China	United States	India	Japan	Brazil
■ Users	641,601,070	279,834,232	243,198,922	109,252,912	107,822,831
■ Rank	1	2	3	4	5

Fig. 1 World top 5 countries' global internet users

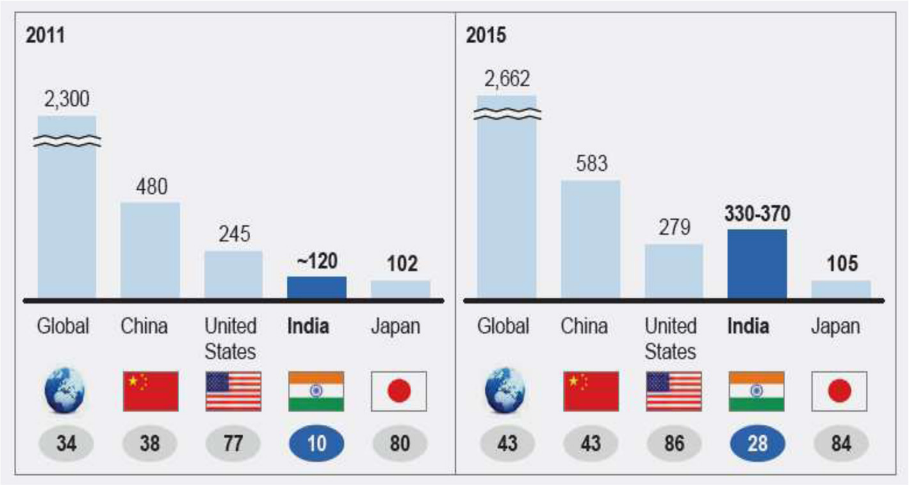


Fig. 2 The projected global internet user's of the world top five countries by 2015

countries. Banking organizations should be advanced in technology adoption to grow up their business services profitably.

Attacks on online banking

The online banking system has tremendous benefits as listed in Table 1, even then why everyone is not using it? The reason is that online banking services could be subjected to the cyber criminals. In [3, 4], authors describe various cyber-attacks that are taking place on different components of online banking, example Spy-Eye malware. Most popular attacks on e-banking are phishing and pharming. These two attacks steal the user login confidential. In [6–10], authors describe various possible solutions to the phishing, Distributed Denial-of-Service, man-in-the browser and cross-site attacks, none of which are suitable for cloud-based environment. LeBlanc.D [15] present various risks associated with e-banking functionalities. PetrHanaeek et al. [24] present a comparative study on e-banking authentications and their attacks.

Table 1 E-banking benefits

To bank	To customer
Per unit operation cost is saved up to 90 %	Transparency and Flexibility
Enables true relationship banking	Self-service banking
Integrate to multi-channel strategies	Personalization
Higher cross-selling opportunities	Time saving
Helps gain customers prompt feedback on products and services	Transactions can be performed at 24 × 7 days with lowest charging rates
	Can gain discount benefits from online purchases

Security in online banking

Authentication is the primary and fundamental operation for many applications, systems and technologies. Till now the traditional authentication with passwords and PIN has dominated the world computing. Present ICT is server-based and it requires stronger authentication methods to provide well enhanced online services. Because of this reason, password and PIN authentication is nearing to the end of their life cycle in many applications. Most of the banks provide user-id, password and also one time password for securing online access to the financial accounts. This type of authentication process is not at all securable because passwords can be obtained by dictionary attack or from specific site information and prior studies as described in [11–14]. Large number of methods and tools are available to compromise passwords. Among which, some of the tools are listed in Table 2 and in [16–18] authors described some other passwords hacking techniques and tools. Traditional authentication is not at all secure for e-banking. So, banks in countries like the USA, Japan, China and other listed in [19] have moved to the biometrics security systems. Specifically, banks in countries like USA and Estonia are using fingerprint biometric security.

Table 2 The password hacking tools

Password hacking tools	Links
John the ripper	www.openwall.com
THC Hydra	www.thc.org
Pwdump	www.foofus.net
Rainbow Crack	www.antsight.com
Brutus	http://sectools.org/tool/brutus/
Cain and Abel	www.oxid.net

Fingerprint biometrics

Biometric fingerprint authentication is the next innovation in banking organizations for providing secure online banking services. Fingerprint security systems are used in broad range of applications for verifying user's identity. Some of the applications are time and attendance system, ATM, PC, laptop and mobile authentication, Aadhaar cards and voter registration. Key advantages of fingerprint security systems are more convenience, highly secure and provides accountability. Seyyede Samine Hosseini et al. [19] conducted a survey on different biometric authentication systems and they found out that more than 47 % of banks are using fingerprint security in their financial dealings. Specifically in Asia 52 % of banking organization are using fingerprint security systems. Because of bio-metric fingerprint accuracy, reliability, scalability, convenience, cost and stability, it is universally accepted for banks in any operational environments [21].

Our trusted framework

In public cloud, protecting user credentials and account details from the cloud service provider and other malicious users is a challenging task. As a result we have investigated a generic trusted framework for protecting user credentials and online banking data in the public cloud. The major components we proposed as a part of trusted framework are fingerprint-based authentication protocol, authorization protocol for validating access rights and protection gateway for preserving data security and privacy in public cloud. We consider that the user authentication credentials, authorization rights and look-up tables are stored and maintained with highly secured in-house databases or trustee databases as shown in Fig. 3.

A consumer who wants to avail the online banking services needs to register at the bank before making any transactions. Along with account details, consumer has to submit his/her personal identification details such as permanent address proof, mobile number and most importantly a fingerprint for registration. In this process, a new idea is proposed where the user can select a user-id (*UID*) and password. We put a restriction that the password must contain at least one digit, one control character, uppercase and lowercase letters and one punctuation symbol is quite strong. In our implementation of this registration phase, we have followed the proper rules and regulations to create, lockout and reset passwords as described in [31–33, 37].

In registration phase, a random secrete key is generated and it is combined with the user biometric fingerprint

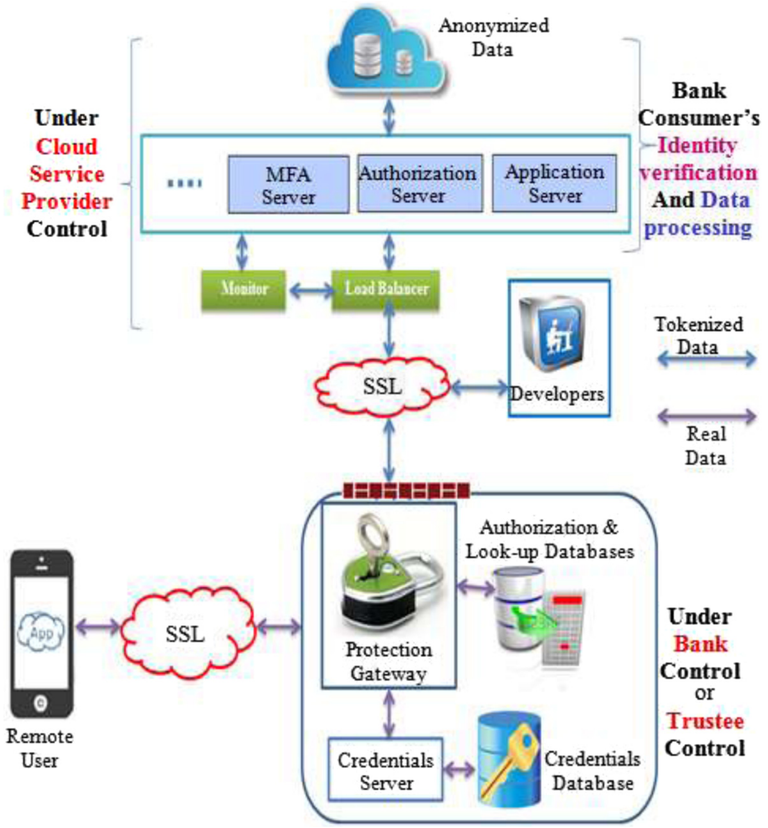


Fig. 3 Proposed model for securing online banking services in the Cloud

data using *exclusive OR* operation. The secrete key is also encrypted using user's biometric fingerprint data. User-id, hashed password, encoded fingerprint data and encrypted secrete key values are stored in the highly secured database. In login phase, Bank Authentication Server (BAS) verifies the user input password matching status, if it is true then it sends the encrypted secrete key to the user registered mobile number through *SMS* for encoding user input biometric fingerprint data. In this process, user decrypts secrete key using his/her biometric fingerprint data and then they encode their fingerprint data for authentication. In authentication phase, BAS and user sends hashed encoded fingerprint data to the Cloud Authentication Server (*CAS*) for authentication. *CAS* checks the hashed password and performs the matching. Our investigated fingerprint-based authentication process is briefly illustrated in Fig. 4.

To describe our authentication approach, we introduce some important terminologies. We denote the registered password as *PWD*, biometric fingerprint data as *B* and the user input password as PWD^* and biometric fingerprint data as B^*. We also indicate the registered hashed password as *h(PWD)* and hashed encoded biometric fingerprint data as *BB*. User input hashed password is denoted as $h(PWD^*)$ and hashed fingerprint data is indicated as BB^*. Further we use Δ as a matching algorithm for checking correctness of the biometric data, and the function δ_k is used with secrete key *k* for encoding biometric fingerprint data using *exclusive OR* operation. The function δ_k cannot be computationally reversible without *k* and will not affect on Δ matching results. The user and bank authentication server send $\{h(PWD^*),\ h(\delta_k(B^*))\}$, $\{h(PWD),\ h(\delta_k(B))\}$ respectively to *CAS* for verification. CAS checks the $h(PWD^*) = h(PWD)$ and matches $\Delta\ (BB,\ BB^*) = (h(\delta_k(B)),\ h(\delta_k(B^*)))$ without storing matching log records locally. Thus, *CAS* cannot learn the user password and fingerprint data.

After successful verification of the status of the user password and fingerprint data matching, Bank Authorization Server (*BARS*) sends *OTP* to the Authorized User (*AU*) through SMS and user access rights and *OTP* to the Cloud Authorization Server (*CARS*). *AU* also sends received *OPT* and transaction details to *CARS*. *CARS* checks *OTP* sent by *AU* and *BARS*. If *OPT* and access rights are matched, then CARS instructs Cloud Application Server (*CAPS*) to perform the user transaction. At the end, *CAPS* and protection gateway carry out following actions:

i. Cloud application server performs the transaction and immediately sends the resultant transaction details to the protection gateway server without storing any data locally.
ii. The transaction values are adjusted in a tokenization database using tokenization knowledge associated with account number and balance.
iii. The protection gateway sends the updated obscured data values to *CAPS*.
iv. *CAPS* stores the obscured data elements rather than real. The obscured values will be used for most of the transactions.

If the authorized user requires the original data, then he/she has to request to the protection gateway. The real data is never stored in any of the cloud databases including log records. Protection gateway highly restricts the access of original data in most of the cases. In this way protection gateway limits the potential exposure to the malicious users.

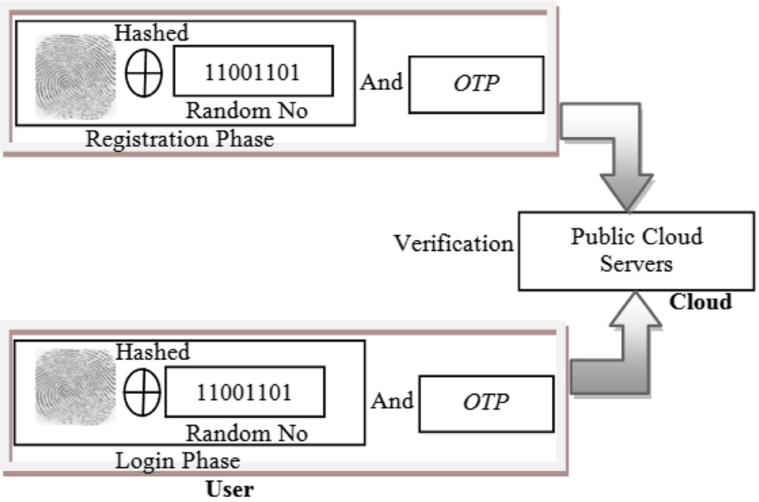


Fig. 4 Proposed verification model for multi-factor authentication

Our protection mechanisms

In this section, we describe in detail functionalities of our protection mechanisms. These mechanisms enable the banking organizations to maintain their own control over the customer sensitive data in public cloud.

Authentication protocol

We present a MFA protocol, in which user's hashed password and fingerprint data verification will be performed using following three phases.

Initialization phase, bank and cloud authentication servers prepare the pair of access keys for remote user authentication. *BAS* prepares a public key indicated as $basp_{bk}$ and a private key denoted as $basp_{rk}$. *CAS* also prepares the public and private key pairs as $casp_{bk,}$ and $casp_{rk,}$. Both the servers keep their private keys as secrete and supplies their public keys to end users.

In registration phase, user registers with bank as follow.

1) A user *U* who wants to avail the banking online services must produce a valid personal identity and mobile number at the bank. Next, the user needs to select a user-id (*UID*) and password and also need to pick a random secrete key '*k*'. User fingerprint will be captured and then client registration module computes *h(PWD)*, $BB = h(\delta_k(B)) = h(k \oplus B)$, and $E_B(k)$, where *h(.)* is a one-way hash function and $E_B(.)$ is symmetric encryption function.
2) Client registration module sends *UID, h(PWD), BB,* and $E_B(k)$) credentials to *BAS* through highly secure *SSL* channel.
3) *BAS* stores all these details and their *status* in a highly secured database as depicted in Table 3, where *status* denotes whether the credentials are registered and unrevoked or not. In our scenario we consider that each server will keep their private keys and all other severs *ID* and public keys for communicating other servers.
4) *BAS* sends registration details and status to the user through highly secure *SSL* channel.

The login and authentication phase takes ten steps for validating correctness of the end user identity as shown in Fig. 5 and described as below. Here, we used five random strings named as *u*, *v*, *w*, *x*, and *y* for encrypting authentication data.

Table 3 The online banking customer credentials

UID	h(PWD)	BB	$E_B(k)$	Status
UID_1	$h(PWD_1)$	BB_1	$E_B(k_1)$	Valid/Invalid
UID_2	$h(PWD_2)$	BB_2	$E_B(k_2)$	Valid/Invalid
….	….	….	….	….
UID_i	$h(PWD_i)$	BB_i	$E_B(k_i)$	Valid/Invalid
….	….	….	….	….

1) User inputs *UID* and PWD^*, and then client module computes $C_0 = E_{casp_{bk}}\left(h(PWD*)\|E_{basp_{bk}}(UID)\|u\right)$ using *CAS* and *BAS* public keys $casp_{bk}$ and $Basp_{bk}$ respectively, where C_0 is a cipher text and *u* is a random string. User then sends C_0 to *CAS.*
2) *CAS* decrypts C_0 using its private key $casp_{rk}$ and then obtains $h(PWD*), E_{basp_{bk}}(UID)$ and *u*. Temporarily it keeps $h(PWD^*)$ and *ue* used for later purpose, and then derives $C_1 = E_{basp_{bk}}\left((CASID)\|E_{basp_{bk}}(UID)\|\right)$ using *BAS* public key $basp_{bk}$ and sends *C1* to *BAS.*
3) *BAS* obtains *CASID* and $E_{basp_{bk}}(UID)$ by decrypting C_1 using $basp_{rk}$ and then decrypts *UID* using same private key. BAS checks *CASID* to ensure that C_1 has come from the proper cloud authentication server and then checks *UID* value in the credential database, if found and valid, then derives $C_2 = E_{casp_{bk}}(BASID\|h(PWD)\|v)$ using *CAS* public key and sends C_2 to *CAS.*
4) *CAS* decrypts C_2 using its private key $casp_{rk}$ and obtains *BASID*, *h(PWD)* and *v* and then checks $h(PWD^*) = h(PWD)$. If both the passwords are equivalent, then *CAS* computes $C_3 = e_v(pwd\ status \parallel w)$ where *pwd status* is the user password verification status. *CAS* sends C_3 to the bank authentication server.
5) *CAS* also derives $C_4 = e_u(CASID \parallel w)$ and sends C_4 to the user.
6) If the password checking status is true, then *BAS* will send Encrypted Secrete Key (*ESK*) i.e., $\varepsilon_B(k)$ to the user mobile through SMS.
7) *BAS* computes $C_5 = E_w(BB \parallel x)$ and sends to *CAS.*
8) User retrieves the secret key *k* by decrypting $\varepsilon_B(k)$ using his/her biometric fingerprint data B^* as $k = A(\varepsilon_B(k), B^*)$, where *A*(.) is an extracting function corresponding to $\varepsilon_B(.)$. The user then computes $BB^* = h(\delta_k(B^*)) = h(k \oplus B^*)$ and derives $C_6 = e_w(BB^* \parallel y)$ and sends C_6 to *CAS.*
9) Finally, cloud authentication server performs the matching function on *C5* and *C6* i.e., $\Delta\ (BB, BB^*) = (h(\delta_k(B)),\ h(\delta_k(B^*))$ and checks the matching score whether it is greater than or equal to a predefined threshold. If it is true, then *CAS* considers the user request legitimate. Next, *CAS* sends fingerprint matched status and the available cloud authorization server *ID (CARSID)* to *BAS* as $C_7 = E_x(fingerprint\ status \parallel CARSID)$.
10) *CAS* also derives $C_8 = E_y(CARSID)$ and sends to the user.

The logically related steps of the login and authentication phase are given in Algorithm 1.

ALGORITHM 1: LOGIN AND AUTHENTICATION PROCESS

```
Input: User Id, password and fingerprint biometrics and
        random secrete keys.
Output: Status of authentication process.
  1)  U     Inputs UID and $PWD^*$
            $C_0 = E_{casp_{bk}}(h(PWD^*)\|E_{basp_{bk}}(UID)\|u)$
            $U \rightarrow CAS$        $C_0$
  2)  CAS   $D_{casp_{rk}}(C_0) = (h(PWD^*)\|E_{basp_{bk}}(UID)\|u)$

            Keeps $h(PWD^*)$ and $u$
            $C_1 = E_{basp_{bk}}((CASID)\|E_{basp_{bk}}(UID))$
            $CAS \rightarrow BAS$      $C_1$
  3)  BAS   $D_{basp_{bk}}(C_1) = ((CASID)\|E_{basp_{bk}}(UID))$
            Checks CASID
            $D_{basp_{br}}(\mathrm{E}_{basp_{bk}}(UID)) = UID$

            Checks UID status, if valid
            $C_2 = E_{casp_{bk}}(BASID\|h(PWD)\|v)$
            $BAS \rightarrow CAS$      $C_2$
  4)  CAS   $D_{casp_{rk}}(C2) = (BASID\|h(PWD)\|v)$
            Checks BASID and $h(PWD^*) = h(PWD)$
            $C_3 = e_v(pwd\ status\|w)$
            $CAS \rightarrow BAS$      $C_3$
  5)  CAS   $C_4 = e_u(CASID\|w)$
            $CAS \rightarrow U$        $C_4$
  6)  BAS   $d_v(C_3) = (pwd\ status\|w)$
            If pwd status is true
            $BAS \rightarrow U$        ESK
  7)  BAS   $C_5 = e_W(BB\|x)$
            $BAS \rightarrow CAS$      $C_5$
  8)  U     $d_u(C_4) = (CASID\|w)$
            $k = \mathcal{A}(\varepsilon_B(k), B^*)$
            $BB^* = h(\delta_k(B^*)) = h(k \oplus B^*)$
            $C_6 = e_W(BB^*\|y)$
            $U \rightarrow CAS$        $C_6$
  9)  CAS   $d_W(C_5) = (BB\|x)$
            $d_W(C_6) = (BB^*\|y)$
            $\Delta(BB, BB^*) = (h(\delta_k(B)), h(\delta_k(B^*))$
            $C7 = e_X(fingerprint\ status\|CARSID)$.
            $CAS \rightarrow BAS$      $C_7$
  10) CAS   $C_8 = e_y(CARSID)$
            $CAS \rightarrow U$        $C_8$
$PWD^*$ : User input password
    $B^*$: User input biometric fingerprint data
    $\mathcal{A}$ : An extraction function
    $\Delta$: Biometric fingerprint matching function
    $\|$: Concatenation operation
$Ep_{bk}(.)$: A public key encryption function with the cloud or
          bank authentication server's public keys
$Dp_{br}(.)$: A decryption function corresponding to $Ep_{bk}(.)$
  $e_{rs}(.)$: A symmetric encryption function with the cloud or
           bank authentication server's random strings as a
           key
  $d_{rs}(.)$: A decryption function corresponding to $e_{rs}(.)$
```

Finally in this phase, bank authentication server decrypts C_7 as d_x (C_7) using its random string x and obtains fingerprint status and the available cloud authorization server *ID*. If the fingerprint *status* is true, then *BAS* will supply *CARSID* to the *BARS* for further communication. The user also decrypts C_8 as d_y (C_8) using his/her random string y and obtains the *CARSID* for further data processing.

Authorization and transaction management protocol

In this sub-section, we describe an authorization and transaction management protocol which provides the session key and data access tokens for each authorized user. As described in the sub-section *4.1*, if a user is legitimate, then the Bank Authorization Server (*BARS*) will generate a unique random number referred as one-time transaction password or session key and access tokens for performing a specific transaction. In [41], we propose an enhanced symmetric *RBAC* which we used for enforcing access policies and managing legitimate users' authorizations.

The control flow of our proposed authorization and transaction management protocol is depicted in Fig. 6 and described below:

1) Bank authorization server sends *OTP* to the user registered mobile number through SMS.
2) *BARS* then computes the cipher text using *CARS* public key as $C_0 = E_{carsp_{bk}}(OTP\|access\,tokens\|BAPSID)$ where *OTP* is the session key, *access tokens* represents the obscured account number, user permissions and fund access constraints. *BAPSID* is the bank application server ID. Bank authorization server sends C_0 to *CARS.*
3) *AU* derives $C_1 = E_{carsp_{bk}}(OTP\|transaction\,details)$ and sends C_1 to *CARS,* where *transaction details* indicates the information regarding fund transfer, paying utility bills, mobile recharge, etc.
4) Cloud authorization server decrypts C_0 and C_1 using its private key $carsp_{rk}$ and then checks *OPT's*, if both are same, then the user is allowed to perform some transaction. Cloud Application Server (*CAPS*) performs the data operations using *access tokens* and immediately sends the resultant values to Bank Application Server (*BAPS*) without storing any type of data on local virtual machine as $C_2 = E_{bapsp_{bk}}$ (*transaction results*$\|$*CAPSID*).
5) Cloud application server also sends the transaction status to *AU.*
6) *BAPS* decrypts C_2 using its private key $bapsp_{rk}$, then the transaction values will be mapped to the real data elements and adjusted in highly secured look-up database using tokenization knowledge associated with a specific account number and balance. After

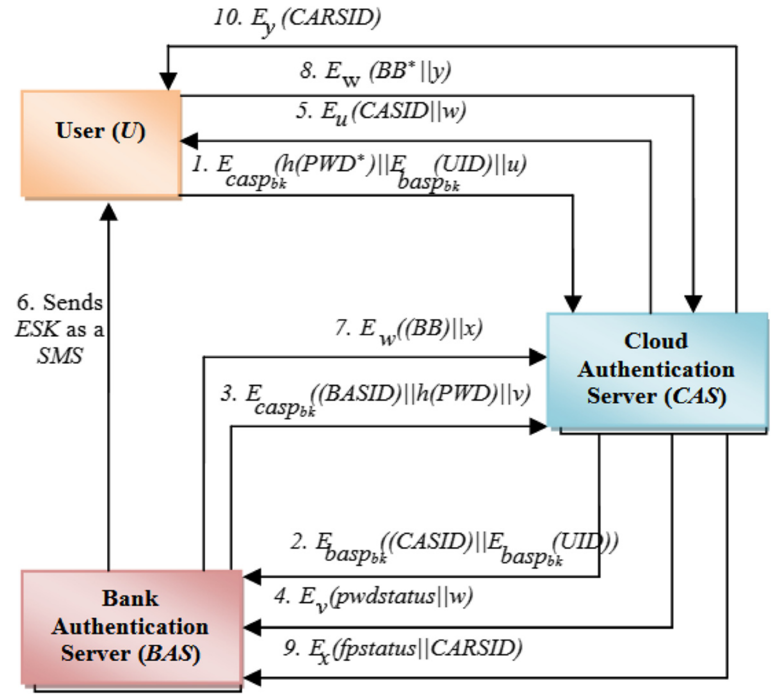

10. E_y (CARSID)
8. E_w (BB^*||y)
5. E_u(CASID||w)
User (U)
1. $E_{casp_{bk}}$($h(PWD^*)$||$E_{basp_{bk}}$(UID)||u)
6. Sends ESK as a SMS
7. E_w((BB)||x)
Cloud Authentication Server (CAS)
3. $E_{casp_{bk}}$((BASID)||h(PWD)||v)
2. $E_{basp_{bk}}$((CASID)||$E_{basp_{bk}}$(UID))
4. E_v(pwdstatus||w)
Bank Authentication Server (BAS)
9. E_x(fpstatus||CARSID)

Fig. 5 Login and authentication protocol

that *BAPS* returns the updated obscured data tokens to the cloud application server. *CAPS* stores the obscured data rather than real sensitive data. The obscured data will be used for most of the transactions. In this protocol, *BARS* and *BAPS* are the part of the protection gateway.

Achieving privacy using data anonymization gateway

In this sub-section, we describe the protection gateway using tokenization techniques. Our proposed gateway is more suitable and effective for preserving privacy of the numerical sensitive attributes than any other privacy methods. In this gateway we use advanced tokenization

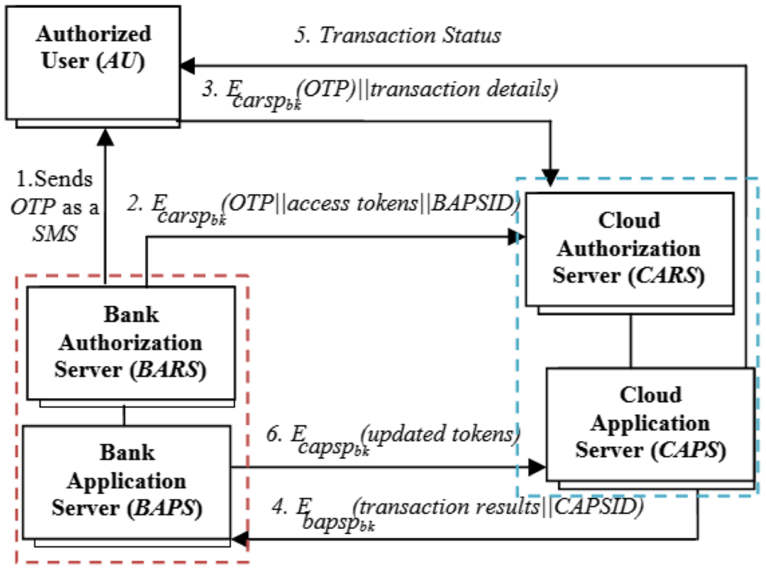

Authorized User (AU)
5. Transaction Status
3. $E_{carsp_{bk}}$(OTP||transaction details)
1.Sends OTP as a SMS
2. $E_{carsp_{bk}}$(OTP||access tokens||BAPSID)
Cloud Authorization Server (CARS)
Bank Authorization Server (BARS)
Cloud Application Server (CAPS)
Bank Application Server (BAPS)
6. $E_{capsp_{bk}}$(updated tokens)
4. $E_{bapsp_{bk}}$(transaction results||CAPSID)

Fig. 6 Authorization and transaction management protocol

techniques for obscuring bank account details. We used *t*-closeness mechanism properties for adding more de-identification to the tokenized data values. Data security and privacy concerns can be eliminated successfully by using our proposed protection gateway. Banking organization's can make use of this protection gateway for preserving data security and privacy in public cloud, while it still allows the authorized users to perform useful transactions.

k-Anonymity

The *k*-anonymity is a popular data de-identification privacy mechanism created by L. Sweeney [41] for data publishing. The main consideration of this scheme is that for every record there should be *k-1* other record has to be exist such that all these records quasi-attribute values should be equal. So, that each and every record is de-identifiable from *k-1* other records. For example, consider the three attributes date of birth, street and city in a record. The record is *k*-anonymized when other *k-1* records have the same date of birth, street and city values. In general, more data privacy can be achieved when we take higher value for *k*. As given in Table 4, k-anonymity divides the record attributes into three categories and assigns appropriate properties and required actions need to be taken.

Figure 7 (i) shows sample bank customer records where an attribute's pin-code, age and gender are considered as *quasi-identifiers* (*QI*) and account balance consider as *sensitive attribute*. An attribute is considered to be *sensitive* when its value in a database should not allow adversary to disclose. The attribute is called quasi-*identifiers* when it is not considered as sensitive. Figure 7(ii) depicts three 4-anonymity customer datasets derived from the Fig. 7(i). Here, '*' indicates suppressed data values, for example pin-code = 4758* means pin-code is in the range of [47580–47589] and the age = 4* means that the age is in between [40–49]. Note that there are 4 records which have the same *QI* values in the Fig. 7(ii) that is why this table is in *4*-anonymity. A *k*-anonymity scheme guarantees that each user record cannot be disclosed from other users' record in a dataset of size *k*. The fictitious tuples would be included in the database if there are no *k* identical *QI*. The effect of fictitious records will be removed on the processing.

l-Diversity

Ashwin M et al. [43] proposed *l*-diversity scheme which enhances the concept of *k*-anonymity. The *k*-anonymity approach cannot resist the homogeneity and background knowledge attacks. In cloud-based applications these two attacks are possible. So, more powerful data de-identification mechanisms are required for achieving adequate data privacy in the public cloud. The use of *l*-diversity will address the weaknesses of *k*-anonymity. By using *l*-diversity, we can achieve more data anonymization. The main difference between these two approaches is that in *k*-anonymity, *k* number of records must have the same values for *QI* and in *l*-diversity; each *quasi-identifiers* dataset must have at least *l* different sensitive values. Figure 7(iii) preserves *3*-diversity for the data given in Fig. 7(ii). While *l*-diversity provides a guarantee of the stronger privacy than *k*-anonymity, the adequate *l*-diversity may not be achieved with natural occurrence of the sensitive values. The fictitious data records need to be added to the given datasets to increase occurrences for preserving *l*-diversity and also need to compensate the effects of these record values when performing some useful computations. Also, in this scheme there is a chance of occurrence of probabilistic inferences.

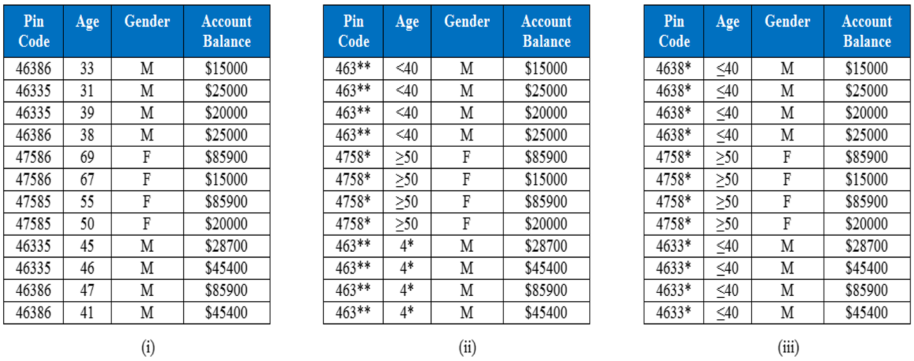
t-Closeness

In [44], a *t*-closeness approach was described, which provides the further enhancement over the *l*-diversity. In this scheme, the authors have taken a specific measurement called *t*-closeness which uses a characteristic that the distance of the sensitive values distribution in the original datasets and generalized datasets must be within threshold *t*. Hence, these two distributions distance can be measured in earth mover distance metric. This approach is more suitable and effective for preserving privacy of the numerical sensitive attributes than any other privacy methods. Because of this advantage we use this scheme in our proposed gateway.

The above schemes' characteristics are developed in our proposed protection gateway to add more anonymization to the online banking account details. Various data obfuscation techniques are available for the above privacy mechanisms to replace the values of *QI*. These techniques are basically categorized as generalization and permutation based data anonymization. The generalization based

Table 4 *K*-anonymity attributes

Attribute type	Example	Action need to be performed	Property
Sensitive	A/c Balance, PIN	Can be de-linked from individual user	Key piece of data that users considers to be sensitive about revealing
Key	Name, Phone No., SSN	Obscure or Removed	Can be derived individual identity directly
Quasi-Identifiers	Gender, Birthday, Pin code	Generalize or Suppress	Attributes used to link with external information for identifying individuals



Pin Code	Age	Gender	Account Balance
46386	33	M	$15000
46335	31	M	$25000
46335	39	M	$20000
46386	38	M	$25000
47586	69	F	$85900
47586	67	F	$15000
47585	55	F	$85900
47585	50	F	$20000
46335	45	M	$28700
46335	46	M	$45400
46386	47	M	$85900
46386	41	M	$45400

(i)

Pin Code	Age	Gender	Account Balance
463**	<40	M	$15000
463**	<40	M	$25000
463**	<40	M	$20000
463**	<40	M	$25000
4758*	≥50	F	$85900
4758*	≥50	F	$15000
4758*	≥50	F	$85900
4758*	≥50	F	$20000
463**	4*	M	$28700
463**	4*	M	$45400
463**	4*	M	$85900
463**	4*	M	$45400

(ii)

Pin Code	Age	Gender	Account Balance
4638*	≤40	M	$15000
4638*	≤40	M	$25000
4638*	≤40	M	$20000
4638*	≤40	M	$25000
4758*	≥50	F	$85900
4758*	≥50	F	$15000
4758*	≥50	F	$85900
4758*	≥50	F	$20000
4633*	≤40	M	$28700
4633*	≤40	M	$45400
4633*	≤40	M	$85900
4633*	≤40	M	$45400

(iii)

Fig. 7 Data anonymization of bank customer records (i) original data (ii) 4-anonymity (iii) 3-diversity

techniques are used to replace the values of *QI* with the suppressed values as shown in the above *k*-anonymity examples. Although generalization works effectively and the association privacy and presence privacy are eliminated, but it has considerable data loss [45]. To address this data loss problem, the permutation based data anonymization techniques have come in existence. The permutation based techniques decompose the sensitive and *QI* values into two different tables without data suppression and keeps the *QI* values table in the published area. Although permutation techniques achieve better performance over the generalization based techniques, it cannot resist the presence privacy, because where the exact values of *QI* are placed in the published environment that enables presence leakage. None of these data anonymization techniques are suitable for achieving desired privacy in the cloud-based environment.

Tokenization

In [35, 36], reported that the tokenization is the best approach for protecting sensitive information in the public cloud environment compared to encryption and any other security mechanisms. Tokenization replaces the sensitive data elements with tokens or surrogate values and also maps back to the real data by making use of a secure enclave or look-up table's. Tokenized data cannot be mathematically reversible because the tokens do not have logical relationship with the original data. Typically, for performing computations on the tokens in various cloud applications, tokenization allows to maintain same data type and length for the tokens as like original data. Tokenization process makes the confidential data useless to anyone except the owner of the data. So, the online banking application can make use of the advantages of tokenization for obscuring customer account details. The obscured data values can be stored in public cloud by following *t*-closeness characteristics, so that the inside and outside malicious users cannot disclose the key pieces of data fields. The anonymized data values can be processed in public cloud servers without bothering about malicious users. Later, the computation results can be mapped to the real data elements in the enterprise tokenization database. Table 5 presents the obfuscation techniques for data tokenization.

Example

Table 6 shows two customers account records and are obscured as follow.

- Names are mapped to a new unique values using permutation.
- Prefix preservation retains the birth year on date of birth.
- Maps the street and city data field values to a new single value
- Phone numbers are shortened by truncating end values retains only area code.
- Replaces the confidential part of the account number with a character '*x*'.
- Account balance values are added to a fixed offset using shift.

In this way, the customer account details will be obscured for protecting individual's sensitive data fields.

Figure 8 shows a theoretical example that helps to understand the secure computing in the public cloud using data anonymization. The objective of this example is to transfer some fund from one account to another, without exposing original account number and balance. Here, the confidential parts of each account number such as branch code and product code is replaced with a character '*x*' and balance is added to a fixed offset i.e., 10,000. Also, some fictitious data records are added to the obscured data.

Look-up database or secure enclave holds the anonymized data values and their associated original values. This database is typically secured on the enterprise/trustee network at highly restricted area. Using anonymized data

Table 5 Tokenization techniques for obscuring bank customers data records

Techniques	Description	Customer name	Date of birth	Street	City	Phone number	Account number	Account balance
		Sample data						
		Alice	17.12.1973	Side hill	Downtown	0418-444-4467	6207693489	$5000
		Bob	24.09.1990	North	Brooklyn	0418-444-6423	3001337388	$10000
Permutation	- Maps each data field value to a distinct new value. - Using secure enclave we can translate new value to original value.	Eva	17.12.1973	Side hill	Downtown	0418-444-4467	6207693489	$5000
		Rob	24.09.1990	North	Brooklyn	0418-444-6423	3001337388	$10000
		Customer name data field values mapped to a distinct new values.						
Prefix-Preserving	- Retains the birth year on date of birth or Replace with a dummy date. - Useful for preserving date field format.	Alice	07.02.1973	Side hill	Downtown	0418-444-4467	6207693489	$5000
		Bob	14.04.1990	North	Brooklyn	0418-444-6423	3001337388	$10000
		The birth year will be preserved, but the date of birth is scrambled.						
Hashing	- Maps different data field values to a new single value. - Useful for translating large amount data values to a new value.	Alice	17.12.1973	8704274623		0418-444-4467	6207693489	$5000
		Bob	24.09.1990	7909231657		0418-444-6423	3001337388	$10000
		Each customer street and city is mapped to a unique new value.						
Truncation (or) Non-disclosure	- Data field values to be shortened by truncating end values. - Useful for tokenizing fields.	Alice	17.12.1973	Side hill	Downtown	0418	6207693489	$5000
		Bob	24.09.1990	North	Brooklyn	0418	3001337388	$10000
		The phone number is shortened, but still preserves the customer location.						
Hiding	- Replaces sensitive value with a character (typically x) or constant value '0'. - Useful for preventing sensitive data fields.	Alice	17.12.1973	Side hill	Downtown	0418-444-4467	xxxxxx3489	$5000
		Bob	24.09.1990	North	Brooklyn	0418-444-6423	xxxxxx7388	$10000
		Except last four digits of an account number, all other digits will be substituted with a character 'x'.						
Shift	- A data field value is added to a fixed offset. - Useful for obscuring data field, while allowing authorized users to perform computation in cloud.	Alice	17.12.1973	Side hill	Downtown	0418-444-4467	6207693489	$15000
		Bob	24.09.1990	North	Brooklyn	0418-444-6423	3001337388	$20000
		A fixed offset $10000 is added to the account balance values.						

values, the fund transfer transaction will be performed in the public cloud without exposing original data values. After successful transaction, the resultant values can be correlated to the real data elements using tokenization knowledge associated with account number and balance. A true fund transfer value can be protected as long as the tokenization knowledge remains confidential. Thus the online banking data can be protected in public cloud.

Completeness of authentication protocol

In this section we analyze the completeness of our proposed authentication protocol using belief logic. Burrows, Abadi, and Needham (BAN) logic [59] is the fundamental and popular belief logic which is widely used to analyse the completeness of various authentication schemes, but this logic has some shortcomings [60]. Gong, Needham, and Yahalom (GNY) logic [61] is the extended version of the BAN logic. We used GNY logic [61] to analyze our multi-factor authentication protocol. First, we describe important terminologies that we use in our belief logic and we re-describe our approach according to the GNY logic. Next, we analyze our goals and finally we report assumptions list.

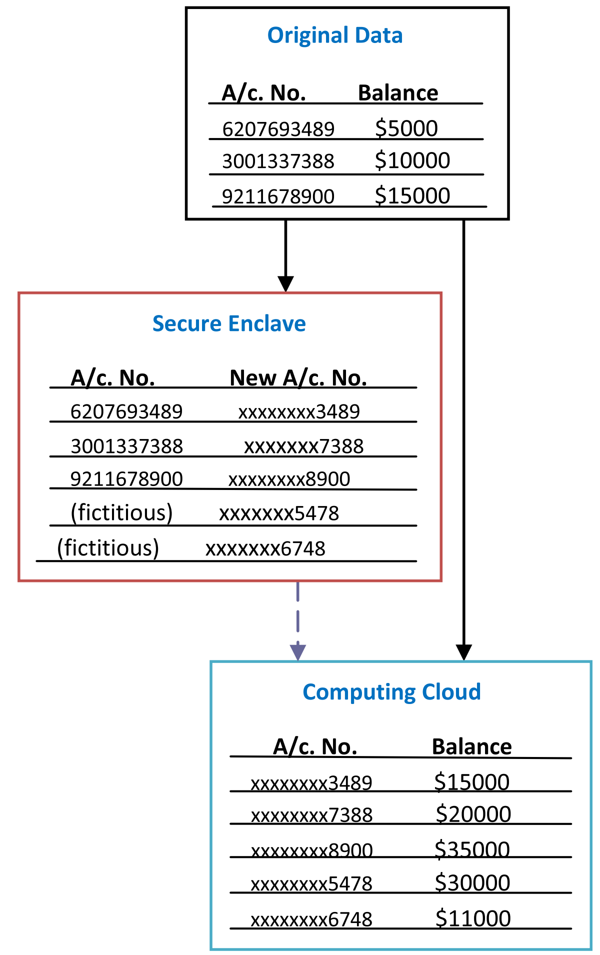
Basic terminologies and statements

In this section we defined key terminologies which we used for analyzing our proposed with GNY logic. Let CP_i and CP_j are the two credential parameters and we introduce the following rationale based on CP_i and CP_j:

- (CP_i, CP_j): conjunction of two rationales $\mathrm{CP_i}$ and $\mathrm{CP_j}$.
- CP_i^*: CP_i is a credential parameter sent by user in login and authentication phase.

Table 6 An example of tokenized bank customer accounts

Customer name	Date of birth	Street	City	Phone number	Account number	Account balance
Eva	07.02.1973	8704274623		0418	xxxxxx3489	$15000
Rob	14.04.1990	7909231657		0418	xxxxxx7388	$20000



Original Data

A/c. No.	Balance
6207693489	$5000
3001337388	$10000
9211678900	$15000

Secure Enclave

A/c. No.	New A/c. No.
6207693489	xxxxxxxx3489
3001337388	xxxxxxx7388
9211678900	xxxxxxxx8900
(fictitious)	xxxxxxx5478
(fictitious)	xxxxxxx6748

Computing Cloud

A/c. No.	Balance
xxxxxxxx3489	$15000
xxxxxxxx7388	$20000
xxxxxxxx8900	$35000
xxxxxxxx5478	$30000
xxxxxxxx6748	$11000

Fig. 8 Secure computing in the public cloud using data anonymization

- $h(CP_i)$: one way hashing function on CP_i.
- $\{CP_i\}_{+k}$, $\{CP_i\}_{-k}$: asymmetric encryption and decryption of CP_i using a public key + k and a private key -k.
- $\{CP_i\}_k$, $\{CP_i\}_k^{-1}$: symmetric encryption and decryption of CP_i using a key k.

In our proposed belief logic, following are the statements which we describe the properties of above rationale. Let E_i and E_j are the two entities which participate in the login and authentication approach.

- $E_i \triangleleft E_j$: E_j is informed E_i
- $E_i \ni CP_i$: E_i has a credential parameter CP_i
- $E_i \sim CP_i$: E_i conveyed CP_i
- $E_i \equiv \#(CP_i)$: E_i persuaded that CP_i is generated from proper entity
- $E_i \equiv \Phi\ (CP_i)$: E_i feels that CP_i is acceptable
- $E_i \equiv E_i \overset{S}{\leftrightarrow} E_j$: E_i persuaded that S is a proper secrete for E_i and E_j
- $E_i \equiv \xrightarrow{+K} E_j$: E_i trusts that's + K is a proper public key for E_j
- $E_i => CP_i$: E_i has authorization over CP_i
- $E_i \triangleleft {}^*E_j$: E_j informed to E_i that he has not sent any messages in present session

Protocol transformation

Below we map our proposed authentication methodology into $E_i \rightarrow E_j$: CP_i form. We also convert some terminologies of our protocol to satisfy the GNY belief logic. Here, the user is denoted as C, cloud authentication server is indicated as S_1 and bank authentication server is represented as S_2.

1) $C \rightarrow S_1$:$\{\{h(PWD^*), \{UID\}_{K2}, u\}_{K1}\}$
2) $S_1 \rightarrow S_2$:$\{\{CASID\}_{K2}, \{UID\}_{K2}\}$
3) $S_2 \rightarrow S_1$:$\{\{BASID, h(PWD), v\}_{K1}\}$
4) $S_1 \rightarrow S_2$:$\{\{pwd\ status, w\}_v\}$
5) $S_1 \rightarrow C$:$\{\{CASID, w\}_u\}$
6) $S_2 \rightarrow C$:$\{ESK\}$
7) $S_2 \rightarrow S_1$:$\{\{BB, x\}_w\}$
8) $C \rightarrow S_1$:$\{\{BB^*, y\}_w\}$
9) $S_1 \rightarrow S_2$:$\{\{fingerprint\ status, CARSID\}_x\}$
10) $S_1 \rightarrow C$:$\{\{CARSID\}_y\}$

In the above transformation K_1 and K_2 are CAS and BAS public keys respectively and u, v, x and y considered as random strings. Here, the client input UID, PWD* and BB* we regard same as BAS database details.

We then converted our protocol into $E_i| \sim CP_i$ and $E_i \triangleleft E_j$as given below. Here, if the rationale CP_i and its terms are appears first time either in $E_i| \sim CP_i$ or $E_i \triangleleft E_j$ then those rationale and terms will be preceded with the star. Our authentication protocol transformation productions are described as follow:

i. $S_1 \triangleleft \{*\{*h(PWD*), \{*UID|\}_{K2}, *u\}_{K1}\}\}\sim$
 $> C|\equiv C \leftrightarrow^{u} S_1$
ii. $S_2 \triangleleft \{*\{\{*CASID\}_{K2},\ \{UID\}_{K2}\}\sim > S_1|\equiv S_1 \leftrightarrow^{K2} S_2$
iii. $S_1 \triangleleft \{*\{*BASID, *h(PWD), *v\}_{K1}\}\sim > S_2|\equiv S_2 \leftrightarrow^{v} S_1$
iv. $S_2 \triangleleft \{*\{*pwd\ status,\ *w\}_v\}\sim > S_1|\equiv S_1 \leftarrow^{w} S_2$
v. $C \triangleleft \{*\{CASID,\ w\}_u\}\sim > S_1|\equiv S_1 \leftrightarrow^{w} C$
vi. $C \triangleleft \{*ESK\}\sim > S_2|\equiv S_2$
vii. $S_1 \triangleleft \{*\{*BB,\ *x\}_w\}\sim > S_2|\equiv S_2 \leftrightarrow^{x} S_1$
viii. $S_1 \triangleleft \{*\{*BB*,\ *y\}_w\}\}\sim > C|\equiv C \leftrightarrow^{y} S_1$
ix. $S_2 \triangleleft \{*\{*fingerprint\ status,\ *CARSID\}_x\}\sim > S_1|\equiv S_1 \leftrightarrow^{x} S_2$
x. $C \triangleleft \{*\{CARSID\}_y\}\sim > S_1|\equiv S_1 \leftrightarrow^{y} C$

Goals

The goals of our proposed belief logic are categorized into four aspects as follow:

1) Message content authentication

In first flow, S_1 feels and believes that the client request is valid and recognizable

$$S_1|\equiv\Phi\{\{h(PWD*),\ \{UID\}_{K2},u\}_{K1}\}.$$

In second flow, S_2 feels and believes that the S_1 request is valid and recognizable

$$S_2|\equiv\Phi\{\{CASID\}_{K2},\{UID\}_{K2}\}.$$

In third flow, S_1 feels and believes that the S_2 response is valid and recognizable

$$S_1|\equiv\Phi\{\{BASID,\ h(PWD),v\}_{K1}\}.$$

In fourth flow, S_2 feels and believes that the S_1 response is valid and recognizable

$$S_2|\equiv\Phi\{\{pwd\ status,\ w\}_v\}.$$

In fifth flow, C feels and believes that the S_1 response is valid and recognizable

$$C|\equiv\Phi\{\{CASID,\ w\}_u\}.$$

In sixth flow, C feels and believes that the S_2 message is valid and recognizable

$$C|\equiv\Phi\{ESK\}.$$

In seventh flow, S_1 feels and believes that the S_2 response is valid and recognizable

$$S_1|\equiv\Phi\{\{BB,\ x\}_w\}.$$

In eighth flow, S_1 feels and believes that the C response is valid and recognizable

$$S_1|\equiv\Phi\{\{BB*,\ y\}_w\}.$$

In ninth flow, S_2 feels and believes that the S_1 response is valid and recognizable

$$S_2|\equiv\Phi\{\{fingerprint\ status,\ CARSID\}_x\}.$$

In tenth flow, C feels and believes that the S_1 response is valid and recognizable

$$C|\equiv\Phi\left\{\{CARSID\}_y\right\}.$$

2) Message origin authentication

In first flow, S_1 believes C originated request

$$S_1\equiv C|\sim\{\{h(PWD*),\ \{UID\}_{K2},u\}_{K1}\}.$$

In second flow, S_2 believes S_1 originated message

$$S_2\equiv S_1|\sim\{\{CASID\}_{K2},\{UID\}_{K2}\}.$$

In third flow, S_1 believes S_2 sent response

$$S_1\equiv S_2|\sim\{\{BASID,\ h(PWD),v\}_{K1}\}.$$

In fourth flow, S_2 believes S_1 conveyed message

$$S_2\equiv S_1|\sim\{\{pwd\ status,\ w\}_v\}.$$

In fifth flow, C believes that S_1 sent the response

$$C\equiv S_1|\sim\{\{CASID,\ w\}_u\}.$$

In sixth flow, C believes S_2 originated encrypted secrete key

$$C\equiv S_2|\sim\{ESK\}.$$

In seventh flow, S1 believes S_2 sent response

$$S_1\equiv S_2|\sim\{\{BB,\ x\}_w\}$$

In eighth flow, S_1 believes C conveyed response

$$S_1\equiv C|\sim\{\{BB*,y\}_w\}.$$

In ninth flow, S_2 believes S_1 sent response

$$C\equiv S_1|\sim\{\{fingerprint\ status\|CARSID\}_x\}.$$

In tenth flow, C believes S_1 conveyed response

$$C\equiv S_1|\sim\{CARSID\}.$$

3) Credentials Verification and Validation

In third flow, S_2 believes and verifies S_1 sent *UID*, if it found and valid, then S_2 sends C's hashed password to S_1, otherwise user authentication process will be terminated.

$$S_2|\equiv S_1\ni\{\mathrm{UID}\}.$$

In fourth flow, S_1 believes and verifies C and S_2 sent hashed passwords, if passwords are matched, then S_1 sends response to the C and $S_{2,}$ otherwise authentication process will be terminated.

$$S_1|\equiv C,\ S_2\ni\{h(PWD),h(PWD*)\}.$$

In ninth flow, S_1 believes and validates C and S_2 sent hashed fingerprints data using biometric matching functions, if fingerprints are matched, then S_1 sends response to the C and $S_{2,}$ otherwise authentication process will be terminated.

$$S_1|\equiv C,\ S_2\ni\{h(\delta_k(B)),\ h(\delta_k(B^*))\}.$$

4) Generation of Session keys

C and S_1 believes that *u* is a one-time session key shared between C and S_1

$$\mathrm{C}|\equiv \mathrm{S}_1 \equiv \mathrm{C} \overset{\mathrm{u}}{\leftrightarrow} \mathrm{S}_1.$$

S_2 and S_1 believes that v is a one-time session key shared between S_2 and S_1

$$\mathrm{S}_2|\equiv \mathrm{S}_1 \equiv \mathrm{S}_2 \overset{v}{\leftrightarrow} \mathrm{S}_1.$$

S_1 and S_2 believes that w is a one-time session key shared between S_1 and S_2

$$\mathrm{S}_1|\equiv \mathrm{S}_2 \equiv \mathrm{S}_1 \overset{\mathrm{w}}{\leftrightarrow} \mathrm{S}_2.$$

S_1 and C believes that w is a one-time session key shared between C and S_1

$$\mathrm{S}_1|\equiv \mathrm{C} \equiv \mathrm{S}_1 \overset{\mathrm{w}}{\leftrightarrow} \mathrm{C}.$$

S_2 and S_1 believes that x is a one-time session key shared between S_2 and S_1

$$\mathrm{S}_2|\equiv \mathrm{S}_1 \equiv \mathrm{S}_2 \overset{\mathrm{x}}{\leftrightarrow} \mathrm{S}_1.$$

C and S_1 believes that y is a one-time session key shared between C and S_3

$$\mathrm{C}|\equiv \mathrm{S}_1 \equiv \mathrm{C} \overset{\mathrm{y}}{\leftrightarrow} \mathrm{S}_1.$$

Assumption list

To analyze our authentication protocol using belief logic we made the following list of assumptions:

- S_1 has public key + K, private key –K and a one-time random string w $\mathrm{S}_1 \ni +\mathrm{K}, \mathrm{S}_1 \ni -\mathrm{K}, \mathrm{S}_1 \ni w$
- S_1 prepared one-time random string w for encrypting session credential details. So that we assume C and S_2 believes w is shared more securely $\mathrm{S}_1|\equiv \mathrm{S}_1 \overset{w}{\Leftrightarrow} \{\mathrm{C},\ \mathrm{S}_2\}$.
- Since w is generated by S_1 in our authentication approach, so that S_1 has w and persuaded that w is fresh, and also assumes that w is used by C and S_2 for encrypting session credential details $\mathrm{S}_1 \ni w,\ \mathrm{S}_1 \equiv \#(w)$.
- C prepared one-time random strings u and x for encrypting session details. We assume that S_1 believes u and x are shared more securely between C and $\mathrm{S}_1\mathrm{C}|\equiv \mathrm{C}_1 \overset{\mathrm{u,x}}{\Leftrightarrow} \mathrm{S}_1$.
- Since the one-time random strings u and x are prepared by C, so that C has u and x, and persuaded that u and x are fresh, and also assumes that u and x are used by S_1 for encrypting session credential details $\mathrm{C} \ni (\mathrm{u},\ \mathrm{x}), \mathrm{C} \equiv \#(u,\ x)$.
- S_2 has a public key + K_1, private key –K_1 and a one-time random strings v and y $\mathrm{S}_2 \ni +\mathrm{K}_1,\ \mathrm{S}_2 \ni -\mathrm{K}_1,\ \mathrm{S}_2 \ni \{\mathrm{v},\ \mathrm{y}\}$.
- S_2 has prepared one-time random strings v and y for encrypting authentication credentials. We assume that S_1 believes v and y are shared more securely between S_2 and $\mathrm{S}_1\mathrm{S}_2|\equiv \mathrm{S}_2 \overset{v,y}{\Leftrightarrow} \mathrm{S}_1$.
- Since the one-time random strings v and y are prepared by S_2, so that S_2 has v and y, and persuaded that v and y are fresh, and also assumes that v and y are used by S_1 for encrypting session credential details $\mathrm{S}_2 \ni (v,\ y), \mathrm{S}_2 \equiv \#(v, y)$.

Logic analysis

By using GNY belief logic we analyzed our authentication protocol and we can also prove that our proposed methodology achieves our objectives. Below we described the logical postulates adoption of our proposed protocol to achieve its objectives, where we taken T_3 and T_4 logical postulates from the GNY logic [61].

1) The first flow:

$$\frac{\mathrm{S}_1 \triangleleft \left\{ \{h(PWD*), \{UID\}_{K1} u\}_{K} \right\}, \mathrm{S}_1 \ni -\mathrm{K},\ \mathrm{C} \ni u,\ \mathrm{C} \equiv \#(u)}{\mathrm{S}_1 \triangleleft \left\{ h(PWD*),\ \{UID\}_{K1,} u \right\}} (\mathrm{T4}).$$

If S_1 is informed by the client C that the message $\{h(PWD^*), \{UID\}_{K1}, u\}_{+K}$ is encrypted with S_1 public key + K, then S_1 obtains $\{h(PWD^*), \{UID\}_{K1}, u\}$ using corresponding private key $–K$. From the received message, S_1 decrypted contents are formulated as

$$\frac{\mathrm{S}_1|\equiv)\Phi\left(h(PWD*),\ \{UID\}_{K1,} u \right), \mathrm{S}_1 \ni -\mathrm{K}}{\mathrm{S}_1|\equiv)\Phi\left\{ \left\{ h(PWD*),\ \{UID\}_{K1,} u \right\} +_{K} \right\}} (\mathrm{R}_1, \mathrm{R}_2).$$

If S_1 private key –K is matched for decryption, then S_1 accepts the client request and believes that the client's $h(PWD^*)$, $\{UID\}_{K1}$ and u are recognizable and considers for further authentication process. Therefore, we can understand that S_1 believes client request and it can be formulated as follow

$$\frac{\mathrm{S}_1|\equiv)\Phi\{\{h(PWD*),\ \{UID\}_{K1}, u\}\} +_{K}, \mathrm{S}_1 \ni +\mathrm{K}}{\mathrm{S}_1 \Big|\equiv \Phi\{h(PWD*),\ \{UID\}_{K1}, u\}} (\mathrm{R}_1,\ \mathrm{R}_3).$$

2) The second flow:

$$\frac{\mathrm{S}_2 \triangleleft \{\{CASID\}_{K1}\{UID\}_{K1}\} \mathrm{S}_2 \ni -\mathrm{K}_1}{\mathrm{S}_2 \triangleleft \{CASID,\ UID\}} (\mathrm{T}_4).$$

If S_2 is informed by the CAS server S_1 that the message $\{\{CASID\}_{K1}, \{UID\}_{K1}\}$ is encrypted with S_2 public key + K_1, then S_2 obtains message contents *CASID* and *UID* using corresponding private key $–K_1$. From the received message, S_2 decrypted contents are formulated as

$$\frac{S_2|\equiv)\Phi(CASID,\ UID), S_2\ni-K_1}{S_2\big|\equiv\Phi\left\{\{CASID\}_{K1}, \{UID\}_{K1}\right\}}(R_1,\ R_2).$$

If S_2 feels *UID* is recognizable, found and valid, then S_2 entitled to believes that the rationale parameters *CASID, UID* are fresh and generates one-time random string v for further communication

$$\frac{S_2|\equiv\Phi)(CASID,\ ID),\ S_2\ni-K_1,\ S_2\ni v,\ S_2\equiv\#(v)}{S_2\big|\equiv\#\left\{\{CASID\}_{K1}, \{UID\}_{K1}\right\}}(F_1,\ F_7).$$

Therefore, S_2 strongly believes that the credential parameters received in the second flow are fresh

$$\frac{S_2\lhd\left(\{CASID\}_{K1},\ \{UID\}_{K1}\right),\ S_2\ni-K_1,\ S_2|\equiv\Phi)(\{CASID,\ UID\})}{S_2\big|\equiv\sim\left\{\{CASID\}_{K1},\ \{UID\}_{K1}\right\}}(I_1).$$

Below given conditions are holds: 1) S_2 receives the rationale {{*CASID*}$_{K1}$, {*UID*}$_{K1}$} that is encrypted with public key + K_1; 2) S_2 believes that all the decrypted credential components are recognizable 3) S_2 entitled to trust that S_1 sent message is fresh. Therefore, S_2 verifies *CASID* and *UID*, if verification is successful, then S_2 believes that the client and S_1 are legitimated entities. Therefore, we can understand that S_2 trusts C and S_1 and continues communication.

According to the proposed belief logic, S_2 believes that the server S_1 is honest. We assumes $S_2| \equiv S_1 = > S_1| \equiv *$ and we form the following logical postulates for further adoption

$$S_2|\equiv)S_1 => S_1|\equiv*),\ S_2|\equiv)\left\{\{CASID\}_{K1},\ \{UID\}_{K1}\right\},\ S_2|\equiv S_1)\sim$$
$$\frac{\left\{\{CASID\}_{K1},\ \{UID\}_{K1}\right\}, S_2\ni-K_1}{S_2|\equiv S_1|\equiv S_2)\overset{K_1}{\leftrightarrow} S_1)}(J_2).$$

3) The third flow:

$$\frac{S_1\lhd\left\{\{BASID,\ h(PWD),\ v\}_{+K}\right\},\ S_1\ni-K,\ \ S_2\ni v,\ S_2\equiv\#(v)}{S_1\lhd\{BASID,\ h(PWD),\ v\}}(T_3).$$

If S_1 is informed by S_2 that the message {*BASID, h(PWD), v*}$_{+K}$ is encrypted with S_1 public key + *K*, then S_1 obtains {*BASID, h(PWD), v*} using corresponding private key *–K*. From the received message, S_1 decrypted contents are formulated as

$$\frac{S_1|\equiv\Phi)(BASID,\ h(PWD),\ v),\ S_1\ni-K}{S_1\big|\equiv\Phi\left\{\{BASID,\ \ h(PWD),\ v\}_{+K}\right\}}(R_1,\ R_2).$$

Below given conditions are holds: 1) S_1 receives the rationale {*BASID, h(PWD), v*}$_{+K}$ that is encrypted with public key + K; 2) S_1 believes that all the decrypted credential components are recognizable 3) S_1 decrypts the rationale {*BASID, h(PWD), v*}$_{+K}$ using private key -K; 4) S_1 trusts that v is fresh one-time random string and used for further communication with S_2; 5) S_1 entitled to trust that S_2 sent message is fresh. Then, S_1 validates the hashed passwords received from C and S_2, if matched, and then S_1 believes that the client is legitimate entity.

Therefore, we can understand that S_1 trusts the client C and continues authentication process. According to the proposed belief logic, S_1 believes that the server S_2 is honest. We assumes $S_1| \equiv S_2 = > S_2| \equiv *$ and we form the following logical postulates for further adoption

$$S_1|\equiv S_2) => S_2|\equiv_*,\ )S_1|\equiv\#\left\{\{BASID,h(PWD),v\}_{+K}\right\},$$
$$\frac{S_1|\equiv S_2\sim\left\{\{BASID,h(PWD),v\}_{+K}\right\}, S_1\ni\text{-}K)}{S_1\equiv S_2|\equiv S_1\leftrightarrow^{v} S_2}(J_2)$$

4) The fourth flow:

$$\frac{S_2\lhd\left\{\{pwd\,statu,\ w\}_v\right\}S_2\ni v}{S_2\lhd\{CASID,\ UID\}}(T_4).$$

If S_2 is informed by the CAS server S_1 that the message {*pwd status, w*}$_v$ is encrypted with one-time random string v, then S_2 obtains {*pwd status, w*} using v. From the received message, S_2 decrypted contents are formulated as

$$\frac{S_2|\equiv)\Phi(pwd\,status,\ w),\ S_2\ni v}{S_2\big|\equiv\Phi\left\{\{pwd\,status,\ w\}_v\right\}}(R_1,\ R_2).$$

If S_2 feels *pwd status* is recognizable and true, then S_2 entitled to believes that the rationale parameters *pwd status* and w are fresh and also it generates one-time random string y

$$\frac{S_2|\equiv\Phi)(pwd\,status,\ w),\ S_2\ni v,\ S_2\ni w,\ S_2\equiv\#(\mathrm{w})}{S_2\big|\equiv\left\{\{pwd\,status,\ w\}_v\right\}}(\ F_1,\ F_7).$$

Therefore, S_2 strongly believes that the parameters received from S_1 are fresh

$$\frac{S_2\lhd *\ \left(\{pwd\,status,\ w\}_v\right),\ S_1,\ S_2\ni v,\ S_2|\equiv\Phi(\{pwd\,status,\ w\}))}{S_2|\equiv S_1\sim\left\{\{pwd\,status,\ w\}_v\right\}}(I_1).$$

Below given conditions are holds: 1) S_2 receives the rationale {*pwd status, w*}$_v$ that is encrypted with one-time random string v; 2) S_2 believes that all the decrypted credential components are recognizable 3) S_2 entitled to trust that S_1 sent message is fresh. Then after, S_2 verifies *pwd status* if it is *true*, then it believes that the client is legitimated entity. Therefore, we can understand that S_2 trusts C and continues authentication communication.

According to the proposed belief logic, S_2 believes that the server S_1 is honest. We assumes $S_2| \equiv S_1 = > S_1| \equiv *$ and we form the following logical postulates for further adoption

$$\mathrm{S}_2|\equiv)\mathrm{S}_1 => \mathrm{S}_1|\equiv_*,)\,\mathrm{S}_2|\equiv\#\left\{\{\mathit{pwd\,status},\ w\}_v\right\},$$
$$\frac{\mathrm{S}_2|\equiv)\mathrm{S}_1\sim\left\{\{\mathit{pwd\,status},\ w\}_v\right\},\ \mathrm{S}_1,\ \mathrm{S}_2\ni v}{\mathrm{S}_2\equiv\mathrm{S}_1|\equiv\mathrm{S}_2\leftrightarrow^{\mathrm{v}}\mathrm{S}_1}(\mathrm{J}_2).$$

5) The fifth flow:

$$\frac{\mathrm{C}\lhd\left\{\{\mathit{CASID},\ w\}_u\right\},\mathrm{C},\ \mathrm{S}_1\ni u}{\mathrm{C}\lhd\{\mathit{CASID},\ w\}}(\mathrm{T}_3).$$

If the client C is informed by S_1 that the message {*CASID, w*} is encrypted with CAS and C shared random string *u*, then C obtains {*CASID, w*} using *u*. From the received message, C decrypted contents are formulated as

$$\frac{\mathrm{C}|\equiv\Phi)(\mathit{CASID},\ w),\ \mathrm{C},\ \mathrm{S}_1\ni u}{\mathrm{C}|\equiv\Phi)\{\mathit{CASID},\ w\}_u}(\mathrm{R}_1,\ \mathrm{R}_2).$$

Below given conditions are holds: 1) C receives the rationale $\{\mathit{CASID},\ w\}_u$ encrypted with one-time random string *u*; 2) C believes that all the credential components received are recognizable 3) C entitled to trust that S_1 sent message is fresh. Therefore, C verifies S_1 identity *CASID* if it is *valid*, then C believes that the S_1 is legitimated entity. Therefore, we can understand that C trusts S_1 and continues communication.

6) The sixth flow:

$$\frac{\mathrm{C}\lhd\{\mathit{ESK}\}}{\mathrm{C}\lhd\{\mathit{ESK}\}}(\mathrm{T}_3).$$

Whenever C receives *ESK* through SMS from S_2, it is entitled to believe that *ESK* is fresh and then decrypts it with clients input fingerprint data and decrypted *ESK* will be used for encoding clients input fingerprint data. Therefore, we can understand that C trusts S_2 and continues communication.

7) The seventh flow:

$$\frac{\mathrm{S}_1\lhd\left\{\{BB,\ x\}_w\right\},\ \mathrm{S}_1,\ \mathrm{S}_2\ni w,\ \mathrm{S}_2\ni x,\ \mathrm{S}_2\equiv\#(\mathrm{x})}{\mathrm{S}_1\lhd\{BB,\ x\}}(\mathrm{T}_3).$$

If S_1 is informed by S_2 that the message $\{BB,\ x\}_w$ is encrypted with one-time random string *w*, then S_1 obtains *BB* and *x* using *w*. From the received message, S_1 decrypted contents can be formulated as

$$\frac{\mathrm{S}_1|\equiv\Phi)(BB,w),\ \mathrm{S}_1,\mathrm{S2}\ni w}{\mathrm{S}_1\big|\equiv\Phi\left\{\{BB,x\}_w\right\}}(\mathrm{R}_1,\ \mathrm{R}_2).$$

Below given conditions are holds: 1) S_1 receives the rationale $\{BB,\ x\}_w$ i.e., encrypted with one-time random string *w*; 2) S_1 believes that all the decrypted credential components are recognizable 3) S_1 decrypts rationale $\{BB,\ x\}_w$ using *w*; 4) S_1 accepts the S_2 response and considers *BB* and *x* for further authentication process. Therefore, we can understand that S_1 believes S_2 message and it can be formulated as follow

$$\frac{\mathrm{S}_1|\equiv\Phi)\left\{\{BB,x\}_w\right\},\ \mathrm{S}_1,\ \mathrm{S}_2\ni w}{\mathrm{S}_1|\equiv\Phi\{BB,x\}}(\mathrm{R}_1,\ \mathrm{R}_3).$$

8) The eighth flow:

$$\frac{\mathrm{S}_1\lhd\left\{\{BB*,\ y\}_w\right\},\ \mathrm{S}_1,\ \mathrm{C}\ni w,\ \mathrm{C}\ni y,\mathrm{C}\equiv(y)}{\mathrm{S}_1\lhd\left\{\{BB*,\ y\}_w\right\}}(\mathrm{T}_4).$$

If S_1 is informed by the client C that the message $\{BB^*, y\}_w$ is encrypted with S_1 one-time random string *w*, then S_1 obtains *BB** and *y* using *w*. From the received message, S_1 decrypted contents are formulated as

$$\frac{\mathrm{S}_1|\equiv\Phi)(BB*,y),\ \mathrm{S}_1\mathrm{C}\ni w}{\mathrm{S}_1\big|\equiv\Phi\left\{\{BB*,y\}_w\right\}}(\mathrm{R}_1,\ \mathrm{R}_2).$$

If S_1 private key –K is matched for decryption, then S_1 accepts the client request and believes that the client's *BB** and *y* are recognizable and considers for further authentication process. Therefore, we can understand that the S_1 believes client request and it formulated as follow

$$\frac{\mathrm{S}_1|\equiv\Phi)\left\{\{BB*,y\}_w\right\},\ \mathrm{S}_1\mathrm{C}\ni w}{\mathrm{S}_1|\equiv\Phi\{BB*,y\}}(\mathrm{R}_1,\ \mathrm{R}_3).$$

9) The ninth flow:

$$\frac{\mathrm{S}_2\lhd\left\{\{\mathit{fingerprint\ \ status},\ \mathit{CARSID}\}_x\right\}\mathrm{S}_2,\ \mathrm{S}_1\ni x}{\mathrm{S}_2\lhd\{\mathit{fingerprint\ \ status},\ \mathit{CARSID}\}}(\mathrm{T}_4).$$

If S_2 is informed by S_1 that the message $\{\mathit{fingerprint\ status},\ \mathit{CARSID}\}_x$ is encrypted with one-time random string *x*, then S_2 obtains *fingerprint status* and *CARSID* using *x*. From the received message, S_2 decrypted contents are formulated as

$$\frac{\mathrm{S}_2|\equiv\Phi)\{\mathit{fingerprint\ \ status},\ \mathit{CARSID}\}\,\mathrm{S}_2,\ \mathrm{S}_1\ni x}{\mathrm{S}_2|\equiv\Phi\{\mathit{fingerprint\ \ status},\ \mathit{CARSID}\}_x\}}(\mathrm{R}_1,\mathrm{R}_2).$$

If S2 feels *fingerprint status* is recognizable and true, then it entitled to believes that the rationale parameters *fingerprintstatus* and *CARSID* are valid

$$\frac{S_2|\equiv\Phi(\mathit{fingerprint\ \ status},\ \ \mathit{CARSID}),\ \ S_2,\ \ S_1)\ni x}{S_2\big|\equiv\#\left\{\{\mathit{fingerprint\ \ status},\ \ \mathit{CARSID}\}_x\right\}}(F_1,\ \ F_7).$$

Therefore, S_2 strongly believes that the parameters received from S_1 are fresh

$$\frac{S_2 \triangleleft * \left(\{fingerprint\ \ status, CARSID\}_x\right), \\ S_1, S_2 \ni x, S_2 | \equiv \Phi(\{fingerprint\ \ status, CARSID\}))}{S_2 \big| \equiv S_1 \sim \left\{\{fingerprint\ \ status, CARSID\}_x\right\}} (I_1).$$

Below given conditions are holds: 1) S_2 receives the rationale {*fingerprint status, CARSID*}$_x$ i.e., encrypted with one-time random string *x*; 2) S_2 believes that all the credential components received are recognizable 3) S_2 entitled to trust that S_1 sent message is fresh. Then after, S_2 verifies *fingerprint status* if it is *true*, then S_2 believes that the client is legitimated entity. Therefore, we can understand that S_2 trusts C and continues communication.

According to the proposed belief logic, S_2 believes that the server S_1 is honest. We assumes $S_2| \equiv S_1 = > S_1| \equiv *$ and we form the following logical postulates for further adoption

$$\frac{S_2 | \equiv S_1 \Longrightarrow S_1 | \equiv) *, S_2 | \equiv \left\{\{fingerprint\ \ status, CARSID\}_x\right\}), \\ S_2 | \equiv S_1 \sim \left\{\{fingerprint\ \ status, CARSID\}_x\right\}, S_1, S_2 \ni x)}{S_2 | \equiv S_1 | \equiv S_2 \leftrightarrow^x S_1} (J_2).$$

10) The tenth flow:

$$\frac{C \triangleleft \left\{\{CARSID\}_y\right\}, C, S_1 \ni y}{C \triangleleft \{CARSID\}} (T_3)$$

If the client C is informed by S_1 that the message {*CARSID*} is encrypted with S_1 and C shared random string *y*, then C obtains *CASID* using *y*. From the received message, C decrypted contents are formulated as

$$\frac{C | \equiv \Phi(CARSID)), C, S_1 \ni y}{C | \equiv \Phi\{CASID\}_y)} (R_1, R_2).$$

Below given conditions are holds: 1) C receives the rationale {*CARSID*}$_y$ encrypted with one-time random string *y*; 2) C believes that all the credential components received are recognizable 3) C entitled to trust that S_1 sent message is fresh. Then after, C verifies S_1 sent *CARSID* if it is *valid*, then C believes that the *CARSID* is legitimated entity. Therefore, we can understand that C trusts cloud authorization server *(CARSID)* and continues authorization process for useful computations.

Experimental evaluation

The objective of this section is to report the feasibility study of our investigated protection mechanisms. Experimental evaluation of our approach is divided into two subsections. The first subsection describes the performance and properties of the multi-factor biometric fingerprint authentication in terms of security, time taken for login and authentication process, etc. The effectiveness of our protection gateway is addressed in the second subsection in terms of time taken for data anonymization and utility metrics. Before presenting the performance evaluation of our proposed work, we present the experimental setup including login, fingerprint and bank customer databases that we used. With an extensive analysis and experiments we show that our proposed framework not only provides the strong authentication and data security, but also achieves the privacy of the sensitive bank account details.

Experimental setup

Setup

We implemented our mechanisms in C#.NET framework using Visual Studio 2010, Windows Communication Foundation (WCF) with Windows Azure Emulator and SQL Server 2008 R1 SP1. We use a machine running with windows 7 64-bits, 4GB RAM, 2.0GHz Intel Core i7 processor, and a fingerprint reader. We use Elliptic Curve Cryptosystem [53] for public-key encryption/ decryption. Tokenization techniques described in Table 5 we used for data obfuscation, *k*-anonymity generalization algorithm [54] we used for generating anonymous tables. We modify the source code of this algorithm to preserve the *l*-diversity and *t*-closeness characteristics and generated data de-identification tables.

Databases

We use four disjoint fingerprint databases (*FDB's*) which are taken from FVC2006 database [55]. The images of each fingerprint database are captured using four different sensors and details are given in Table 7 with cooperation of 150 heterogeneous participants includes industrial, academic and elderly people. Each *FDB* contains 150 fingers and in-depth 12 samples per finger (i.e., 150 × 12 = 1800). Samples are of exaggerated distortion, dry/wet impressions and large amount of displacement and rotations. Each *FDB* is divided into two disjoint sub-databases as follow:

1. FDB1-A, FDB2-A, FDB3-A, and FDB4-A, where each sub-database stores 140 fingerprint samples.
2. FDB1-B, FDB2-B, FDB3-B, and FDB4-B, where each sub-database stores ten very difficult fingerprint samples.

Table 7 Details of sensors used for capturing databases

Data sets	Sensor type	Resolution	Image size
DS1	Optical	569 dpi	400 × 560 (224 Kpixels)
DS2	Electric Field	250 dpi	96 × 96 (9 Kpixels)
DS3	Thermal sweeping	500 dpi	400 × 560 (200 Kpixels)
DS4	SFinGe v3.0	500 dpi	288 × 384 (108 Kpixels)

Where, B sub-databases contain the most difficult fingerprint images used for evaluating protection strength of the proposed scheme. We generated a login database of size 150 *UID*'s and *PWD*'s, and an adult bank database of size 450000 records using GNU-licensed open source data generator tool [56].

Performance of our login and authentication protocol

First we compare our scheme with password-based and other biometric authentications in terms of computational cost. Next, we illustrate the performance of our fingerprint-based authentication mechanism.

In general, the traditional password-based authentication is more computationally effective than the fingerprint-based authentication, because additional computation power is required for validating biometric fingerprint samples. To develop the multi-factor authentication with fingerprint biometric in a more practical way, the fingerprint samples related computations should be accurate and take less time. In [22], authors pointed out that the practical requirements satisfaction of the biometric fingerprint is more than other types of biometrics (e.g., iris, face, etc.) in terms of authentication and extraction (e.g., fingerprint recognition included in laptop, ATM's etc.).

Comparisons

We took Elliptic Curve Cryptosystem [53] for public-key encryption/decryption and it takes only one modular multiplication for encryption. In our authentication approach, each user requires one symmetric encryption/decryption, one modular multiplication, one exclusive-or and two hash operations in the login and authentication process. Compared to the solutions described in [27, 39, 46], our solution require only two modular exponentiations for each user. In our protocol, a new idea is proposed where a user is allowed to select a user-id (*UID*) and password, not decided by the bank server, so that the user can memorize his/her *UID* and password easily. In [27, 39, 46] authors have used timestamps for authentication. These authentications require clock synchronization between the user and the server computers, and also the login message transmission delay is also limited.

Our approach is used the random nonce values to eliminate the transmission and clock synchronization delay time. Our proposed authentication framework not only performs the credentials validation in *CAS*, but also provides the login and authentication credentials privacy. In [27, 39, 46] checks the credentials in the smart card and does not consider the privacy. Wenyi Liu et al. [48] match the credentials in server and no privacy is provided. Our scheme also does not require any credentials database at the cloud side. Table 8 provides the performance comparisons of our approach with other mechanisms.

Table 8 Performance comparison

	C1	C2	C3	C4	C5
A.Jyoti Choudhury et al. [46]	NO	NO	NO	NO	YES
Wenyi Liu et al. [48]	YES	YES	YES	NO	NO
J. K. Lee et al. [27]	NO	NO	NO	NO	YES
Y.Lee and T. Kwon [39]	NO	NO	NO	NO	YES
Our approach	YES	YES	YES	YES	YES

C1: Requires low computation cost
C2: The user is allowed to select a user-id (*UID*) and password, not decided by the bank server
C3: The clock synchronization is not required between the user and server computers
C4: Not only performs the credentials validation in *CAS*, but also provides the login and authentication credentials privacy
C5: Don't require any credentials database at authentication server's to validate user identity

To the best of our knowledge, our approach is the first multi-factor biometric fingerprint authentication approach which provides biometric fingerprint security and privacy in a cloud-based environment.

Results

We evaluated our proposed authentication protocol using series of experiments with combination of 150 *UID*'s and *PWD*'s, and four fingerprint databases each contains 150 finger images and 12 in-depth samples per finger (i.e., $150(UID\ and\ PWD) \times 12(\text{samples per finger}) = 1800$ credential records in each database). We set a time window bound in minutes for validating user login and authentication credentials in terms of False Negative Rate (FNR) and False Positive Rate (FPR). FNR means the rate of input credentials matched correctly and calculated as $t_p/(t_p + f_n)$, where f_n is false negative and t_p is true positive. *FPR* means the rate of input credentials matched incorrectly and computed as $t_n/(t_n + f_p)$, where t_n is considered as true negative and f_p taken as false positive. The recognition performance of our proposed approach for FVC2006 databases is reported in Fig. 4, where x-axis indicates databases DB1, DB2, DB3, and DB4, and y-axis indicates FNR and FPR percentages. We have set four different time window bounds such as 5, 10, 16 and 20 min for each database and we find out recognition rates (Fig. 9).

We also find out the Rejection Enrollment (RE), Rejection Matching (RM), Average Enrollment Time (AET), Average Matching Time (AMT), Equal Error Rate (EER) and Revised EER (REER) over the FVC2006 databases as shown in Table 9. We consider EER as a unit of measure of fingerprint recognition performance and it denotes where FNR and FPR are equal. The average EER of our mechanism for the FVC2006 databases is 1.44 %. From the Table 4 we can understand that the EER little varies for each input fingerprint database of different sensor type. For example, FDB4 has more equal error rate (i.e., 1.66 %)

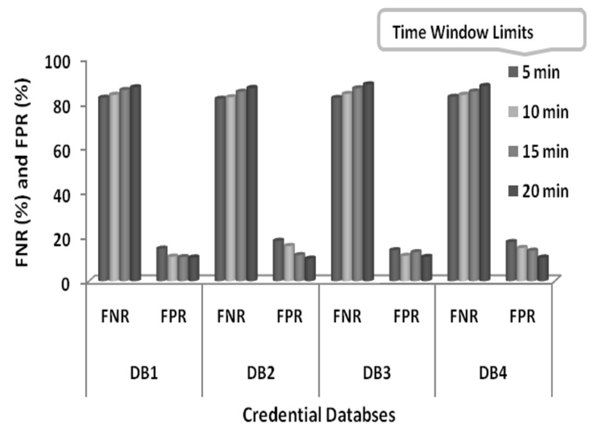


Fig. 9 Performance of our proposed multi-factor authentication

when compared to FDB1 EER value (i.e., 1.15 %) because these two databases differ in resolution and image sizes. Our scheme generated better equal error rate than the existing fingerprint-based works.

Security and privacy analysis

From the above comparisons and results, we show that our MFA protocol is secured and provides the credentials privacy from the malicious users. Here, we maintain the anonymous credentials and access key in the highly secured bank authentication server and matching is performed in the cloud authentication server, so that the real login and the authentication data are not revealed to cloud and bank. We use the random strings and *ESK* for mutual authentication of the users and servers; hence the malicious users cannot pretend to be the authorized users and impersonated requests can be eliminated.

Effectiveness of our protection gateway

The objective here is to assess the effectiveness of our privacy preservation gateway using an adult customer's database of size 450000 records. Table 10 summarizes the description of database in terms of unique number of values, tokenization techniques and generalization hierarchy height we used for each attribute. Here, the account balance is considered as sensitive attribute and others are non-sensitive. A *QI* of size *i* consists of three or more non-sensitive attributes from the database as shown in Table 10. We evaluate the running time taken by our proposed protection gateway to generate the *k*-anonymity ($k = 6$), entropy *l*-diversity ($l = 6$), recursive (c, *l*)-diversity (4, 6) and *t*-closeness ($t = 0.15$ & 0.2) anonymous data tables for varied sizes of quasi-identifiers and are reported in Fig. 10. The running times taken for generating anonymous data tables are similar.

Table 9 Performance of our approach on the four fvc2006 databases

Data base	EER	REER	RE	RM	AET	AMT
FDB1	1.15 %	1.15 %	0.00 %	0.00 %	1.23 s	0.18 s
FDB2	1.49 %	1.49 %	0.00 %	0.00 %	1.53 s	0.19 s
FDB3	1.48 %	1.48 %	0.00 %	0.00 %	1.74 s	0.14 s
FDB4	1.66 %	1.66 %	0.00 %	0.00 %	1.76 s	0.21 s
Avg.	1.44 %	1.44 %	0.00 %	0.00 %	1.56 s	0.18

Table 10 Description of the bank customer database

Attribute	Distinct values in each domain	Tokenization technique	Ht.
Name	112843	Permutation	4
Gender	2	Mixing	1
Birth date	76544	Prefix-preservation	3
Street	93724	Hashing	4
City	45665	Hashing	4
Country	41	Hashing	2
Phone no.	380543	Truncation	4
Occupation	71	Mixing	4
Account no.	450000	Hiding	4
A/c balance	35724	Shift (Sensitive Att.)	

We quantify the utility of our protection gateway in terms of generalization height, minimum average size of the blocks, and *discernibility*. The generalization height [57] is the metric that can be defined as the number of generalization steps performed by an anonymization algorithm while generating anonymous tables. The second metric that implemented as a part of the anonymization algorithm is an average size of the blocks to maintain the anonymity among the data records. The *discernibility* cost metric [58] quantifies the indistinguishable data records from each other. Figure 11 provides the utility experimental results of our protection gateway. In Fig. 11 (i), we found that the minimum height of utility metric is not an ideal, because it generate larger block sizes for the tables with small height. For larger values of height

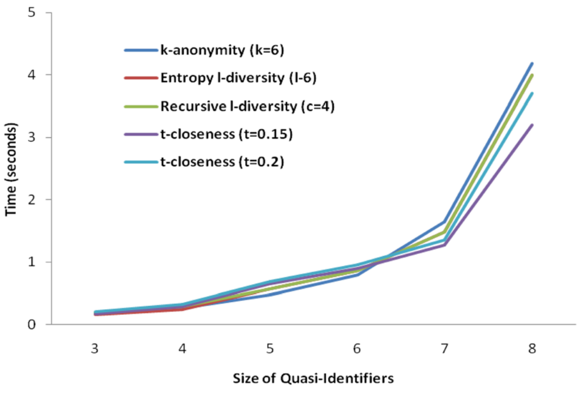
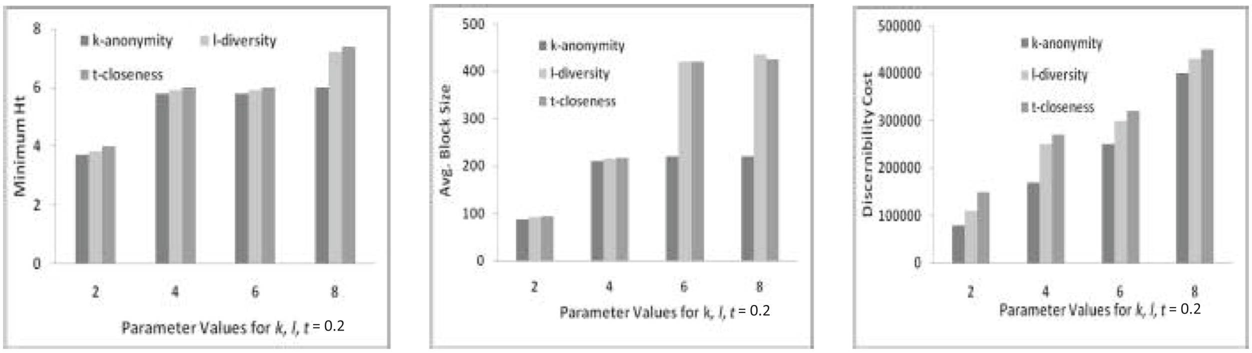


Fig. 10 Performance of proposed protection gateway



(i) (ii) (iii)

Fig. 11 Efficiency of our protection gateway (i) generalization height (ii) average size of blocks (iii) discernibility cost

it produces high quality of anonymizations. For the smaller values of k, (4, l) = 2, 4, 6, 8 and $t = 0.2$, yields higher utility block sizes and discernibility cost as depicted in Fig. 11 (ii) and (iii).

Related work

Developing an efficient trusted framework for public cloud-based environment is an open problem. In the recent years efforts are being made to develop a trusted cloud environment. We divided the existing works into two parts. First part presents the various traditional and cloud-based authentication works. In second part, data security and privacy related works are addressed.

Several traditional multi-factor authentication approaches have been designed to integrate the fingerprint biometrics with smart-card and /or password authentication. Lee et al. [27] developed a user identity verification approach through smart cards, where the registered user supplies his/her password and biometric fingerprint samples in login process. In this scheme password table is not required, but fingerprint and smart-card tables are required for validating user's identities. However, this mechanism was broken by the approaches described in [28, 29]. C. C. Chang et al. [28] reported that Lee's authentication approach cannot resist the conspiring attack. C. H. Lin et al. [29] also discovered an improved scheme that maps the password and fingerprint data into a super password. However this approach cannot resist an impersonation attack. In [38] Yoon et al. presented a solution to resist this attack. This improved solution was broken by Lee et al. in the work done in [39] and they made further enhancement in this scheme. This solution is not broken till now, but it has failed to check the server side biometrics. A multi-factor authentication privacy preserving protocol is proposed by Bhargav et al. [40] using multi-factors namely password, a random string and a fingerprint. In this scheme they formed a cryptographic key by using multi-factors for identity verification. The problem with this scheme is in authentication phase each user needs to find expensive modular exponential computations. However, the above mentioned traditional multi-factor authentication mechanisms are not suitable for cloud-based environment and the approaches described in [27–29, 38, 39] are also not consider the privacy of the user credentials.

Some cloud-based authentication mechanisms are developed in recent years for validating the user identities. A.J. Choudhury et al. [46] presented an authentication framework to integrate user *ID* and password with smartcard. This scheme is not suitable for the public cloud environment, because smartcards adoption is very difficult process and their validation process is easily compromise to the cyber attacks. Rohitash Kumar B et al. [47] proposed a *MFA* framework using *OTP* and *IMEI* number as authentication secrets. In [48], Wenyi Liu et al. described a multi-factor cloud authentication approach using user password and secure user profile. However, the schemes described in [47, 48] are not suitable to achieve our problems described in the *sub-section 1.1*. To address our problems, the user credentials should not be revealed to the cloud service provider, even to the enterprise, because the fingerprint biometric data may also be used for some other applications. Therefore, our authentication approach protects the user credentials from cloud malicious insiders and outsiders.

Data security and privacy remains to be top and legitimate concerns for adoption of public cloud. It is an active and challenging area for researchers to provide efficient solutions. As a result, several data anonymization and privacy preserving schemes are developed to ensure data privacy and security in a public cloud. R. Chow et al. [49] designed a scheme to build in-house cloud by avoiding external cloud. The advantage of this approach is to retain the private/hybrid cloud and to eliminate the public cloud concerns. However, this solution is not affordable and costly for most of the organizations.

Hui Wang [50] presented a privacy-preservation solution using *Ambiguity* and *PriView* methods. The author protected the association and presence leakages by dividing the database into multiple tables using lossless joins, but the problem with this scheme is that still there is an association leakage and information loss, most importantly author has not followed proper referential integrity constraints. K. Puttaswamy et al. [51] discovered another alternative approach for data protection, where they used cryptographic techniques to encrypt all the sensitive data without limiting the application functionalities in the cloud. However, this scheme is slow and the applicability is also limited, because the authors assume that the raw data is not required for web applications, which is a rare case.

Our protection gateway is an extension of the work proposed by Vanessa Ayala-Rivera et al. [52], in which data is anonymized using substitution techniques. These techniques cannot provide high-level protection to the sensitive data in public cloud. Therefore, their framework requires advanced data obfuscation methods and efficient privacy preservation mechanisms.

Conclusion and future work

An adoption of the online banking into cloud will provide expertise solutions, high processing speed, reliable storage and advanced business features at nominal cost. Data security and privacy, residency and legal regulatory laws remain to be top and legitimate concerns preventing the banking organizations from adopting public cloud environment. In this article we described two practical protection mechanisms, the multi-factor biometric fingerprint authentication and protection gateway, which enables the banking organizations to maintain their own controls over the customer sensitive data in a public cloud. Especially, the user credentials and customer account details will not be revealed to the cloud service provider and other malicious users. MFA is used to verify whether the user is authenticated or not to the online banking services. In our approach, fingerprint data is a key factor for authentication. We described MFA protocol using data extraction, biometric matching, and symmetric and asymmetric encryption/decryption algorithms. We also analyze the completeness of our proposed authentication protocol using GNY belief logic.

Our proposed protection gateway allows the enterprises to protect their customer's sensitive information destined for the public cloud and achieves the data privacy concerns. We implemented advanced tokenization techniques and data anonymization mechanisms as integral part of the protection gateway for preserving the privacy of the key piece of information from the inside and outside malicious attackers. Our proposed protection mechanisms make the banking online services more secure and reachable to a common man in cloud. In future work, we are planning to implement query auditing techniques for detecting and preventing the disclosures of the sensitive information and also planning to develop an efficient self-learning algorithm for identifying sensitive data fields in the dynamic cloud datasets.

Competing interests

The authors Mr. Sabout Nagaraju and Dr. Latha Parthiban declare that they have no competing interests.

About the authors

Sabout Nagaraju is currently working as assistant professor in Pondicherry University. He is graduated from G.Pulla Reddy Engineering College, Kurnool and did his post graduation at NIT, Calicut. He is pursuing his Ph.D. from Pondicherry University. His professional experience spans over 8 years in various Engineering colleges and software industry. His areas of interest include Internet of Things, Cloud Computing, Design and Analysis of Algorithms, Cryptography. He has published five papers in international journals and seven papers in national/ international conferences.

Dr. Latha Parthiban has obtained her B.E in Electronics and Communication Engineering from University of Madras, M.E. in Computer Science and Engineering from Anna University, Chennai and has obtained her PhD from Pondicherry University. Her teaching experience spans over 21 years in various Engineering colleges and her research interest includes Soft computing, Expert systems, Image Processing and cloud computing. She has published research papers in 26 international journals and presented papers in 22 international and national conferences. She has also published a book in the area of computer aided diagnosis.

Authors' contributions

Mr. Sabout Nagaraju has made substantial contributions to conception, design, implementation, acquisition of test data, and performed experimental evaluation. Dr. Latha Parthiban has involved in revising it critically for important intellectual content, supervision of the research work and has given final approval of the version to be published. Both authors read and approved the final manuscript.

Acknowledgements

Thanks to the Pondicherry University administration for providing required hardware and software resources to carry out this work successfully.

Received: 12 June 2015 Accepted: 29 October 2015
Published online: 01 December 2015

References

1. Habib S (2012) Internet banking in India consumer concerns and bank marketing strategies. Proc Res J Manag Sci 1(3):20–24
2. Online banking gross transaction volume (GMV) in China from 2008 to 2018 (in trillion yuan), www.statista.com/statistics/248967/online-banking-transaction-volume-in-china/.
3. Ronchi C, Khodjanov A, Mahkamov M, Zakhidov S (2011) Security, privacy and efficiency of internet banking transactions, Proceedings of the 2011 World Congress on. Date of Conference., pp 21–23
4. Hole KJ, Moen V, Tjøstheim T (2006) Case study: online banking security, Presented at IEEE Computer Society., pp 14–20
5. Online banking, http://en.wikipedia.org/wiki/Online_banking.
6. Larcom G, Elbirt AJ (2006) Living with technology: gone phishing, Published in IEEE Technology and Society Magazine., pp 52–55
7. Zhan J, Thomas L (2011) Phishing detection using stochistic learning-based weak estimators, Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium., pp 55–59
8. Ranjan S, Knightly E (2008) High performance distributed denial-of-ServiceResilient Web cluster architecture, Proceedings of the IEEE Network Operations and Management Symposium, 2008., pp 1019–1024
9. Bin Mat Nor F, Jalil KA, Manan JLA (2012) An enhanced remote authentication scheme to mitigate Man-in-the-browser attacks, Proceedings of the 2012 IEEE International Conference., pp 271–275
10. Kerschbaum F (2007) Simple cross-site attack prevention, Proceedings of the Third International Conference., pp 464–472
11. Highland H (1992) Random bits and bytes: testing a password system. Comput Secur 11(2):110–113

12. Klein D (1990) Foiling the cracker: a survey of, and improvements to, password security, Proceedings of the 2nd USENIX UNIX Security Workshop., pp 5–14
13. Morris R, Thompson K (1979) Password security: a case history. Commun ACM 22(11):594–597
14. Spafford E (1992) Observing reusable password choices, Proceedings of the 3rd UNIX Security Symposium., pp 299–312
15. LeBlanc D (2012) Risk perception of internet-related activities, Proceedings of the 2012 Tenth Annual International Conference., pp 88–95
16. Veir M (2009) Password cracking using probabilistic context-free grammars, Proceedings of the 30th IEEE Symposium on Security and Privacy., pp 391–405
17. Theoharoulis K (2009) HighEnd reconfigurable systems for fast Windows' password cracking, Proceedings of the 17th IEEE Symposium onField Programmable Custom Computing Machines., pp 287–290
18. Murakami T (2010) An implementation and its evaluation of password cracking tool parallelized on GPGPU, Proceedings of the International Symposium on Communications and Information Technologies (ISCIT)., pp 534–538
19. Hosseini SS, Mohammadi S (2012) Review banking on biometric in the World's bank and introducing a biometric model for Iran's banking system. J Basic Appl Sci Res 2(9):9152–9160
20. Singh J (2012) Scenario of e-banking in today's life: a survey, Proceedings of the International Journal of Computing & Business Research., pp 1–12
21. Tripathi KP (2011) A comparative study of biometric technologies withReference to HumanInterface, Proceedings of the International Journal of Computer Applications., pp 10–15, http://www.ijcaonline.org/archives/volume14/number5/1842-2493
22. Uludag U, Pankanti S, Prabhakar S, Jain AK (2004) Biometric cryptosystems: issues and challenges. Proc IEEE Spec Issue Multimedia Secur Digit Rights Manag 92(6):948–960
23. Schlich B (2012) The customertakes control: global consumer banking survey 2012., pp 1–64, www.ey.com/globalconsumerbankingsurvey
24. Hanaeek P, Malinka K, Jiri S (2010) e-Banking security -a comparative study, Proceedings of the IEEE A&E Systems Magazine., pp 29–34
25. Core Banking In The Cloud Banking Software Solutions Vendor http://www.hostgeni.net/docs/pdf-core-banking-itecban/.
26. Suresh MC (2010) Cloud computing: strategic considerations for banking & financial services institutions, TCS White Papers., pp 1–24
27. Lee JK, Ryu SR, Yoo KY (2002) Fingerprint-based remote user authentication scheme using smart cards. Electron Lett 38(12):554–555
28. Chang CC, Lin IC (2004) Remarks on fingerprint-based remote user authentication scheme using smart cards. ACM SIGOPS Oper Syst Rev 38(4):91–96
29. Lin CH, Lai YY (2004) A flexible biometrics remote user authentication scheme. Comput Stand Interfaces 27(1):19–23
30. Gnanasambandam C, Madgavkar A, Kaka N (2012) Online and Upcomming: The Internet Impacts on India. McKinsey & Company, New York, pp 1–66
31. The Ultimate Guide for Creating Strong Passwords http://www.thegeekstuff.com/2008/06/the-ultimate-guide-for-creating-strong-passwords/.
32. Create strong passwords https://www.microsoft.com/security/pc-security/password-checker.aspx.
33. Six rules for safer financial transactions online http://www.microsoft.com/security/online-privacy/finances-rules.aspx.
34. 1 in 4 Internet Users Access Banking Sites Globally by Adam Lella, http://www.comscore.com/Insights/Data-Mine/1-in-4-Internet-Users-Access-Banking-Sites-Globally.
35. Cipher cloud information protection overview, White Paper, www.ciphercloud.com, pp.1-14, 2013.
36. Insight into payment security: Encryptiona & Tokenization, A Whitepaper by NVISH Commerce, pp.1-8, website: www.nvish.com.
37. Overcoming Security, Privacy & Compliance Concerns, White Paper, www.ciphercloud.com, pp.1-13, 2013.
38. Yoon EJ, Yoo KY (2005) A new efficient fingerprint-based remote user authentication scheme for multimedia systems. In: 9th Int. Conf. Knowledge-based & intelligent information & engineering systems (KES 2005)., pp 332–338, Paper LNAI 3683
39. Lee Y, Kwon T (2006) An improved fingerprint-based remote user authenticationscheme using smart cards. Proc ICCSA 3981:915–922, Lecture Notes in Computer Science
40. Bhargav-Spantzel A, Squicciarini AC, Bertino E, Modi S, Young M, Elliott SJ (2007) Privacy preserving multi-factor authentication with biometrics. J Comput Secur 15(5):529–560
41. Nagaraju S, Parthiban L, Santhosh Kumar B (2013) An enhanced symmetric Role-Based Access Control using fingerprint biometrics for cloud governance. PCCR 1:12–18
42. Sweeney L (2002) k-Anonymity: a model for protecting privacy. Int J Uncertainty Fuzziness Knowledge Based Syst 10(5):557–570
43. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M (2006) ℓ-Diversity: privacy beyond *k*-Anonymity, ICDE., pp 1–12
44. Li N, Li T, Venkatasubramanian S (2007) t-Closeness: orivacy beyond k-anonymity and l-diversity, ICDE Conference
45. Aggarwal CC (2005) On *k*-anonymity and the curse of dimensionality, Proc. Very Large Data Base Conference (VLDB), Trondheim, Norway., pp 901–909
46. Choudhury AJ, Kumar P, Sain M, Lim H, Jae-Lee H (2001) A strong user authentication framework for cloud computing, 2011 IEEE Asia -Pacific Services Computing Conference., pp 110–115
47. Kumar Banyal R, Jain P, Kumar Jain V (2013) Multi-factor authentication framework for cloud computing, 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation., pp 105–110
48. Liu W, Selcuk Uluagac A, Beyah R (2014) MACA: a privacy-preserving multi-factor cloud authentication system utilizing Big data, 2014 IEEE INFOCOM Workshops: 2014 IEEE INFOCOM Workshop on Security and Privacy in Big Data., pp 518–523
49. Chow R, Golle P, Jakobsson M (2009) Controlling data in the cloud: outsourcing computation without outsourcing control, ACM Workshop on Cloud Computing Security. ACM, Chicago, IL
50. Wang H (2010) Privacy-preserving data sharing in cloud computing. J Comput Sci Tech 25(3):401–414
51. Puttaswamy K, Kruegel C, Zhao B (2011) Silverline: toward data confidentiality in storage-intensive cloud applications
52. Ayala-Rivera V, Nowak D, McDonagh P (2013) Protecting organizational data confidentiality in the cloud using a high-performance anonymization engine., pp 1–8
53. Miller V (1985) Uses of elliptic curves in cryptography, Advances in Cryptology—Crypto85, ser. Lecture Notes in Computer Science, no. 218., pp 417–426
54. Xiao X, Tao Y (2006) Anatomy, "Simple and effective privacy preservation", Proc. Very Large Data Base Conference (VLDB), Seoul, Korea., pp 139–150, The code is taken from http://www.vldb.org/conf/2006/p139-xiao.pdf
55. Cappelli R, Ferrara M, Franco A, Maltoni D (2007) Fingerprint verification competition 2006. Biom Technol Today 15(7-8):7–9
56. GEDIS studio online test data generator http://www.data-generator.com/.
57. LeFevre K, DeWitt D, Ramakrishnan R (2005) Incognito: efficient fulldomain k-anonymity, SIGMOD
58. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases, VLDB
59. Burrows M, Abadi M, Needham R (1989) A logic of authentication. ACM Trans Comput Syst 23(5):1–13
60. Nessett DM (1990) A critique of the Burrows, Abadi, and Needham logic. Oper Syst Rev 24(2):35–38
61. Gong L, Needham R, Yahalom R (1990) Reasoning about belief in cryptographic protocols, Proc 1990 IEEE Computer Society Symp Research in Security and Privacy., pp 234–246