A Novel Multiserver Authentication Scheme Using Proxy Resignature With Scalability and Strong User Anonymity

Hu Xiong, Zhiqing Kang, Jinhao Chen, Junyi Tao, Chen Yuan, and Saru Kumari

***Abstract*—As the indispensable part of the security demand in a distributed system, a variety of online services need to be protected from unauthorized access. Multiserver authentication (MA) is one of the most effective and promising approaches to guarantee one party the authenticity of its partner in the distributed system. In this article, we first elaborate the problem of scalability for MA protocol. After that, a two-level framework for multiserver environment is proposed to convert single-server authentication to MA with provable security and perfect scalability. Also, we give an instance of our novel infrastructure by adopting the idea of proxy resignaturetoachievestronganonymity.Furthermore,thesecurity analysis and performance evaluation show that our protocol is secure and practical.**

***Index Terms*—Anonymity, multiple server, provable security, scalability, three-factor authentication.**

SYMBOL NOTATIONS

Notation	Definition
U_i	User.
S_j	User's "mother" server.
S_k	User's "foreign" server.
RC	Registration center.
SC_i	User's smart card.
ID_i	User's identity.
PW_i	User's password.
$H_i(\cdot)$	One-way hash function.
$\mathrm{Enc}_k/\mathrm{Dec}_k(\cdot)$	Symmetrical en/decryption by *k*.
$\|\|$	String concatenation operator.
$\oplus$	Exclusive-or operation.
SK	Session key.

I. INTRODUCTION

WITH the pervasiveness and advance of mobile Internet, the popularity of mobile devices, i.e., smart phones or laptops, is increasing rapidly [1]. The powerful mobile devices along with advanced mobile Internet technologies give rise to a greatnumberofmobileusersandalargewideofmobileservices such as e-commerce [2], e-medical/health [3], e-voting [4], e-learning [5], and so on. Given the openness, heterogeneousness, and complexity of mobile networks, it is essential for the remote server to check the legitimacy of users to avoid unauthorized usages of online services or resources. Remote authenticationisidentifiedasoneofthemostcommonsolutionstof acilitate secure communication in mobile networks. In remote authentication, a server is allowed to authenticate the remote users and establish a session key used for further communication.

In 1981, the remote authentication protocol was initially introduced by Lamport [6] such that users can be authenticated by theserverbasedontheiridentitiesandcorrespondingpasswords. In this kind of password-based remote authentication, a verifier table containing the passwords or hashed passwords of all users' needs to be maintained by the server. Recent incidents of password leakage [7]–[9] demonstrate that the verifier table stored in theservermaybecompromisedbythehackereffortlessly,which means that the authentication system is broken. To improve the security of password-based remote authentication, smart device (also known as smart card) is introduced as the second factor in the remote authentication. This kind of remote authentication is usually referred to as two-factor authentication, which means that only the user who owns both factors (i.e., smart device and password) is able to log in the server. Despite two-factor authentication protocols provide stronger security, the smart card might be lost or stolen, which lead to a stolen smart card attack [10]. In this case, biometric, as the third authentication factor, is introduced into remote authentication, termed three-factor authentication [11]–[18].

However, the conventional authentication protocols for the single-server architecture suffer a significant shortcoming. If the usersneedtoaccessresourcesprovidedbydifferentservers,they have to register separately with these service providers (SPs). It is extremely difficult for users to manage numerous credentials (i.e., passwords, smart card, and biometric information) corresponding to different servers after *multiple registrations*. To reduce memory burdens, users may use the same password

Manuscript received December 23, 2019; revised March 20, 2020; accepted March 20, 2020. This work was supported in part by the National Science Foundation of China under Grant U1936101, in part by the 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China under Grant MMJJ20170204, the Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, in part by the Sichuan Science and Technology Project under Grant 2018KZ0007, and in part by the Researchers Supporting Project number (RSP-2019/12), King Saud University, Riyadh, Saudi Arabia. *(Corresponding author: Saru Kumari.)*

Hu Xiong, Zhiqing Kang, Jinhao Chen, Junyi Tao, and Chen Yuan are with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: xionghu.uestc@gmail.com; kangzhiqingcn@gmail.com; jinhaochen.cloud@gmail.com; junyitao@usc.edu; chyuan@buffalo.edu). Saru Kumari is with the Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India (e-mail: saryusiirohi@gmail.com).

Digital Object Identifier 10.1109/JSYST.2020.2983198

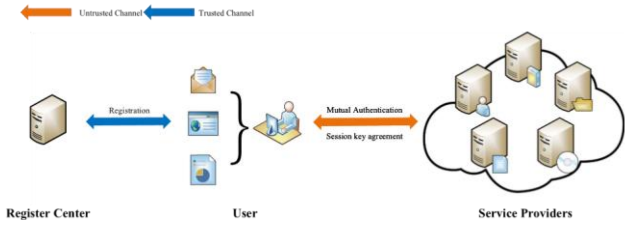


Fig. 1. System model of MA protocol.

during the registration with different servers. It certainly increases the risk of information leakage as the corruption of one server will probably affect the security of other servers, such as the "CSDN" incident. To overcome the drawbacks of the traditional single-server architecture, the multiserver architecture has been introduced to achieve the goal of *single registration* for multiple servers (see Fig. 1): every user is able to visit all SPs after only one registration on the register center (RC). Because of the potential of simplifying the registration process, multiserver authentication (MA) has become increasingly attractive to industry and academia [19]–[22].

So far, most of the existing MA protocols are designed in an *ad hoc* manner such that these solutions are almost cleverly crafted *from scratch*. That is to say, the fruitful achievements in the two-factor authentication for a single server has not been utilized in the designing of MA protocols. By considering that modular design may be meaningful to illustrate a comprehensibleframework,itisdesirabletoconstructmodularMAprotocols that achieve conceptual simplicity and reasonable efficiency. Another problem is that, the efficiency on the user side and server side will be degraded in case large users or servers join the system. However, the scalability of MA protocols has been absent until recently.

In this article, by utilizing the concept of modular design, we propose a novel infrastructure of the MA to overcome the limitations of the previous works by combining the two-factor authentication scheme for the single-server architecture and public-key cryptosystems. Our major contributions can be summarized as follows.

1) We elaborate on the definition of the scalability and the causes why most of existing schemes fail to achieve it in Section II. Concretely, we offer a brief analysis of several schemes to testify that.
2) Next, we formulate the novel two-level infrastructure for multiserver environment, which consists of single-server architecture and public-key cryptosystems, which is able to convert a single-server authentication protocol to a MA protocol. Then, to instantiate our framework, we build a secure two-factor authentication protocol by elaborately incorporating the proxy resignature (PRS) and two-factor authentication.Theproposedprotocolachievessingleregistration, mutual authentication, two-factor security, forward secrecy, and also provides strong user untraceability and scalability.
3) The simulation experiments indicate that the novel protocol for the first time alleviates the computation and storage burdens on the RC. Regarding the computation cost on the users' side and servers' side, our protocol also performs well. In this case, we adopt the bilinear pairing to guarantee the strong/absolute user anonymity so that the communication cost is little higher. However, if this strong/absolute user anonymity is not in the need, we can discard/abandon this high consumption operation.

Inthesecondsection,wewillpresentsomesecurityproperties and elaborate on the scalability. We then formulate our twolevel framework in Section III. In Section IV, we propose our MA protocol, whose performance evaluation is presented in VI. Section VII concludes this article.

II. Evaluation Criteria for MA Protocols

In this part, we are going to give a detailed description of the exiting attacks that harm to the MA protocols. After that, taking into account the practice and security, we further list the dominant criteria that determine the performance of the MA protocols.

A. Attack Model

In order to elaborate on the security requirement of a MA protocol deployed in the real world, various known attacks that may threaten the MA protocols are described as below in detail.

1) Smart card loss attack: Such a type of attack happens as a user's smart card is inadvertently leaking, which is possible in the practical application of the MA protocol. A robust MA protocol is supposed to guarantee the secrecy of a user's personal information (i.e., ID and password) even if his smart card is lost.
2) Insider attack: It is a threat that comes from the user who has access to the servers. The main damage caused by this attack is that the user's identity or the password will be leaked to the adversary.
3) Offline password guess attack: The smart card that involves the personal information of the user is stored locally. In this case, a malicious adversary is able to enumerate all the possible passwords of a user once he obtains a user's smart card.
4) Replay attack: It is carried out by an adversary who desires to access the target session by utilizing the session information generated before.
5) Modification attack: During the phase of the login and authentication, an adversary may illegally eavesdrops and modifies the transmitted message.
6) Server impersonation attack: This type of attack is carried out by an adversary who pretends to be a legitimate server.
7) The man-in-the-middle attack: It is a type of attack that performed by an adversary who makes an endeavor to masquerade as an intermediate who relay and snoop the

secret information transmitted between the legal users and servers.

B. Evaluation Criteria

Compared with single-server authentication, apart from a server and a user, a MA protocol has an additional participant: registration center (RC). Though the participants have some difference, the common purpose of the single-server authentication and the MA is to achieve mutual authentication and key agreement. In a way, it is natural to borrow some ideas from the evaluation criteria of single-server authentication to design security properties of MA. To allow schemes to be rated across a common spectrum, Wang and Wang [23] and [24] have done a remarkable work about the evaluation criteria of two-factor authentication in single-server environment. Apart from that, Liao *et al.* [25] have discussed security vulnerabilities for multiserver environment. Furthermore, a large number of security requirements of MA protocol have been proposed [20], [21], and [26]–[28]. Based on the existing work, a MA protocol shall satisfy the following security properties.

C1. Single registration: To provide convenience and security, the user only needs to register once and then the user can access all the authorized servers in the system.

C2. No online registration center: To reduce the computation overhead, RC only provide registration service with users and servers but need not be online to participate in mutual authentication.

C3. Mutual authentication: To ensure the eligibility of the participants, users and servers should authenticate each other.

C4. Session key agreement: To guarantee the security of the following communication, users and servers should reach an agreement on the session keys, which are used to encrypt the transmitted messages.

C5. Two/Three-factor security: To protect the private key, the private key managed by RC cannot be derived by adversaries even legal users and servers.

*C6.Forwardsecrecy:*Toguaranteethesecurityoftheprevious communication, even if an adversary gets a session key or privatekeysoftheparticipants,he/shecannotderivetheprevious session keys from all the past session keys.

C7. Scalability: To provide openness and practicability, MA protocol should provide perfect scalability in the following two cases.

1) When a new user has registered at RC, servers can provide services to the new user successfully without updating the identities table of users in the RC and all the authenticated servers timely.
2) When a new server has registered at RC, the existing users can successfully perform login and authentication with this server, but the user does not need to waste computation on the extra updating or the interaction with RC.

C8. No verifier table: To protect the users' secret information and reduce the storage overhead, neither a server nor RC need to maintain a database about users' identities and passwords.

C9. No password exposure: To preserve users' passwords, the privileged administrator of the server and RC cannot derive users' passwords by the users information stored in their database and messages interacted with them.

*C10.Useranonymity:*Toguaranteeusers'privacy,thescheme can protect user identity and prevent user activities from being traced.

C11. Password friendly: To provide user-friendliness, users can choose identities and passwords freely, which might be memorable and change passwords locally.

C12. Sound repairability: To provide users with convenient revocation, the smart card revocation should be of good repairability, like a user can revoke the smart card using the previous identity.

C13. No clock synchronization: To adapt to network delays, the participants of the scheme do not need to synchronize their clock time.

C14. Resistance to known attacks: To provide resilience to various attacks, the scheme should resist a variety of prevalent attacks listed in the attack model.

Actually, besides the goal of mutual authentication and key agreement, the scalability that is ignored in multiserver environment all the time is also crucial. It indicates that the system must be extensible for users and servers. That is to say, when more users and servers take part in, the system still runs well, rather than suffering from costly computation overhead due to the join of new users and servers. Considering the exponentially increasing number of the users as well as various services in the real-life scenarios, scalability now is becoming more and more important. For making a clear description, in this part, the discussion of the scalability is divided into two cases: that on the user side and the server side. Obviously, the perfect scalability requires the high extensibility both on the user side as well as the server side.

At first, coming to scalability on the users side, it means that as new users take part in the system, the server is capable of authenticating without frequent update neither on the servers side nor RC side. Specifically, some schemes [29]–[36] are designed like that in an authentication phase, after receiving messagesfromtheusers,theserverwillvalidateusers'identities by users' registration table received from RC, which consists of every user's identity and some corresponding information. To authenticate these new users, the server must update the identities table by RC frequently. Due to the dramatic increase of the users, it is a considerable computation and storage cost for both servers and RC.

Then, it comes to servers' side, with the addition of new servers, the eligible users already existed in the system are able to perform login and authentication to access new services but not need any extra update or interact with RC. Unfortunately, plenty of MA protocols, such as [29]–[31] and [37]–[39], cannot achievethescalabilityontheservers'side.Forexample,in2013, Pippal *at al.* [29] proposed a two-factor authentication scheme for MA. In Pippal's scheme, it is assumed to be "*k*" servers there. RC generates *k* random numbers $(r_1,r_2,r_3,...,r_k)$, and

$$s_j \ (s_j = g^{\prod_{i=1,i\neq j} r_i} \mod N^{k}$$

computes secret key) for each server $S_{j(1\leq j\leq k)}$, which are stored in users' smart card to access services.However,Li*etal.*[40]pointedoutthatPippal'sscheme is not scalable, because when a new server has registered at RC and is already to provide services, the system parameters should be reinitialized and the user has to reregister to the RC. Otherwise, the user cannot access the new services. As a result, Pippal'sschemeisnotscalableontheservers'side.Actually,the inherent limitation in Pippal's scheme is that the authentication between users and servers relies on the servers' secret key s_j, which will be invalid when the *k* increases.

Inaddition,weintroduceanotherprevalentsituationaboutthe lackofscalabilityontheservers'side.Inthissituation,aprotocol is not extensible on the servers' side because of a varying combining parameter stored in the smart card of the user. Take Lin *et al.*'s protocol [37], for example. In Lin's protocol, RC computes $r_j = h(\mathrm{SID}_j||x)$ for server S_j, where x is the master key of RC. In addition, in the user registration phase, RC computes a combining parameter $C_{ij} = E_{h(\mathrm{UID}_i||BIO_i)}[h(h(\mathrm{UID}_i||x)||r_j)]$ and stores it in the smart card, which consists of user's identity UID_i, user's biometric BIO_i, RCs master key x, and server's secret key r_j. r_j in C_{ij} is regarded as user's private key to protect his/her secret information and r_j varies with different servers as well as the combining parameter C_{ij}. In the authentication phase, user U_i uses the C_{ij} to authenticate with server S_j. Accordingly, when a new server S_k is employed by the system, to perform authenticationwithS_k,theuserhavetosubmitsomeinformation about his/her secret information to RC in order to obtain the C_{ik}, termed reregistration. Conclusively, to protect user's secret parameterssuchasidentity,password,orbiometric,aprivatekey will combine with users secret parameters by a hash function or XOR operation and stored in a user's smart card. At the same time, in order to authenticate with the server, this private key is closely connected to the server's secret parameters, even they might be as the same. Accordingly, when the system deploys a new server, the users cannot authenticate with the new server without the crucial combining parameters. It is not reasonable for users to spend extra computation to update by themselves or communicating with RC. Actually, the updating process ought to be transparent for users so that we can truly achieve the goal of single registration for all the services.

Some MA protocols seem to satisfy the requirement of scalability. However, these protocol sacrifice security to scalability. Sometimes the designer of MA protocol chooses the same private keys for users, such as [28] and [40]–[45]. As in [45], all the servers share the same secret key, and all the private keys of users are the same as the servers' secret keys. And in the multiserver environment, it is always believed that only the RC is absolutely trusted and users or servers might be malicious adversaries. In this context, it is the same shared key of users and servers that leads to impersonation attack even offline password attackbyamaliciousserver.Therefore,thesamesharedkeymay achieve the salability perfectly for the private key of the user do not need to update, but it contributes to the security leakage.

In our novel infrastructure for multiserver environment, we resolve the conflict between security and scalability by constructing a two-level authentication architecture and introducing the idea of public-key cryptosystems. On the users side, it is scalable, because the users are located in the first level, which is the single-server architecture and performs the registration, login, and authentication with the "mother" server or under the control of the "mother" server. For another thing, the scalability ontheservers'sideisnotaproblemanymore,becauseitdepends on the mutual trust between servers in the second level.

III. Generic Framework From Single-Server Authentication to MA

In this section, we will elaborate on the two-level framework of MA. In this novel infrastructure, users only need to register at the "mother" server and authenticate the "foreign" servers with the help of the "mother" server. In our design idea, there are two levels in our multiserver architecture. The first level is the single-server architecture, which consists of the users and their "mother" server. The second level is the trust chain among user, "mother" server, and "foreign" server. To establish the trust chain, it is imperative to apply the public-key cryptosystems in our design.

A. Single-Server Authentication

Generally, single-server authentication is divided into four phase.

1) Single-Server Server Registration: There are a public key PK and a private key SK generated for the server (denoted by S). PK is published in the system and SK is kept by S secretly.

2) Single-Server User Registration: The user (denoted by U) chooses identity ID and password PW freely (sometimes inputs biometric feature, we omit this factor in this section). Then the user registers at S by sending information about ID and PW. After computing by S, the user receives a smart card SC. This phase is formulated as

$$U[\mathrm{ID,PW}] \xleftrightarrow{\mathrm{Single-Server-}U\mathrm{-Reg}} S[\mathrm{SK,PK}] \rightarrow \mathrm{SC}.$$

3) Single-Server Login and Authentication: In this phase, the user uses his/her ID, PW, and SC to login successfully. Then the user and the server interact with each other to output a session key sk used for the following communication. This phase is formulated as

$$U[\mathrm{ID,PW,SC}] \xleftrightarrow{\mathrm{Single-Server-Log-Auth}} S[\mathrm{SK,PK}] \rightarrow \mathrm{sk}.$$

4) Single-Server Password Updating: After being authenticated by SC successfully, the user is able to change his/her PW locally and update the corresponding information in SC.

B. Generic Framework for MA

Now, we will describe our two-level infrastructure in detail. In this framework, the biometric is omitted, and the design of the three-factor authentication can be referred to [16] and [46].

1) Multiserver Registration: This process consists of server registration and user registration. Each server will generate a public key PK and a private key SK under the supervision of an authorized institution, i.e., the certification authority (CA), the government. PK is published in the system and SK is kept by the server secretly. The user (denoted by U_i) will select a serverinthesystem ashis/her“mother” serverand registeratthe server. And the rest of the servers in the system are considered as$U_i{}'$s “foreign” server. Actually, this phase acts the same as single-server user registration, besides users and servers possess some different parameters. The details are as follow.

a) Server S_m(“mother” server) has a public key PK_m and a private key SK_m. Another server S_n(“foreign” server) generates PK_n and SK_n.The other servers areas the same.

b) U_i chooses identity UID_i and PW as

$$U_i[\mathrm{UID}_i, \mathrm{PW}] \xleftrightarrow{\text{Multi}-\text{Server}-U-\text{Reg}} S_m[\mathrm{SK}_m, \mathrm{PK}_m]$$

$$\rightarrow [\mathrm{SK}_i,\mathrm{PK}_i]\&\mathrm{Data}_1\&\mathrm{Data}_2$$

where SK_i is the private key for U_i and PK_i is the public key for U_i. Data_1 is a parameter computed by$U_i{}'$s “mother” server by using some information about U_i and its own information to assist U_i to login and authenticate with the “foreign” server in the future, so S_m will store Data_1 in its secret database. Then, S_m stores (SK_i,PK_i) into SC and sends it to U_i.

c) Upon receiving the SC, U_i computes some secret parameters about UID_i, PW, and SK_i, denoted by Data_2 to protect UID_i, PW, and SK_i. Finally, U_i stores Data_2 into SC and SK_i will be removed.

2) Multiserver Login and Authentication With “Mother" Server: U_i inserts SC into a card reader and extracts Data_2 from SC. Then, UID_i, PW, and Data_2 are used for authentication with S_m. This is denoted by

$$U[\mathrm{UID}_i,\mathrm{PW},\mathrm{SC}(\mathrm{Data}_2)]\ \text{Single}\leftarrow\text{------------------}\text{Server}-\text{Log}-\text{Auth}\rightarrow$$

$$S_m[\mathrm{SK}_m,\mathrm{PK}_m] \rightarrow \mathrm{sk}.$$

3) Multiserver Login and Authentication With “Foreign” Server: In this process, U_i will authenticate with S_n with the help of S_m. This is denoted by

$$U[\mathrm{ID},\mathrm{PW},\mathrm{SC}(\mathrm{Data}_2)]$$

$$S_m\text{Multi}\leftarrow\text{------------------}[\mathrm{SK}_{-m\text{Server}},\mathrm{PK}_{-\text{Log}m},-\mathrm{Data}_{\text{Auth}\rightarrow 1}]$$

$$S_n[\mathrm{SK}_n,\mathrm{PK}_n] \rightarrow \mathrm{sk}.$$

a) U_i uses UID_i and PW to perform the first process of authentication with S_m, which is to confirm U_is validity and ensure the messages from U_i are not be tampered with.Onlybypassingthisfirststepcanthefollowingsteps continue.

b) Then, S_m forwards the information about U_i containing $S_m{}'$ s signature to S_n. After receiving these messages, S_n verifies them to ensure validity of U_i and his/her message, and later authenticates with U_i to create a session key sk if the verification successes.

4) MultiserverPasswordUpdating: U_i issupposedtoupdate his/her PW with no need to change UID_i locally.

a) U_i inputs PW and SC will verify whether it is correct.

b) If passing the verification, U_i inputs new password PW and SC updates the information correspondingly.

Obviously,ourgenericframeworkconsiststwolevels:thefirst is single-server authentication and the second level is the trust chain among user, the “mother” server and “foreign” server. When the user authenticates with the “mother” server, it just acts like single-server authentication. And with the help of the “mother” server, the user is able to perform authentication with the “foreign” server. Compared with the traditional framework, this infrastructure enables the existing single-server authentication protocols to convert to MA protocols. As we have discussed in Section II, there is already a “private key” for user protected by PW to authenticate with servers in common single-server authenticationprotocols.So,wenoticethatthereisanadditional public key corresponding to the private key for the user which is the most different from the ordinary protocols of single-server authentication.Thatistoestablishthetrustchainamongtheuser, “mother” server and “foreign” server, which relies on the application of public-key cryptosystem. When converting a singleserver protocol authentication, designers just need to get the user’s private key and public key involved in the process of user authentication. What is more, by adopting this infrastructure, it is easier to achieve strong user anonymity. This is because we divide the authentication into two processes. First, the “mother” server verifies the user. If the user is legitimate, it continues to the second process. The “foreign” server authenticates the “mother”server.Asaresult,itformsatrustchain:user-“mother” server-“foreign” server and the “foreign” server knows nothing about the user’s identity but confirming the user’s legality. In addition, as what has been discussed in Section II, the scalability in this infrastructure is achieved perfectly. Every user in the system has his/her “mother” server and servers in the system authenticate each other by using public-key cryptography system. The “Mother” server is responsible for the authentication between the user and his/her “foreign” servers. So there is no need to worry about a new user knows nothing about the “foreign” servers, which he/she wants to access and vice versa. Also because users only register at one server and submit some secret information, the server is responsible for keeping its “children” users’ secret information safe. Unlike the tradition multiserver architecture in which only RC can be trusted, all the servers in our infrastructure can be trusted in some way, since a server only possess its "children" users secret information and these users' information are also only exposed to their mother server. Once information leakage happens, the server will be blamed. By this means, malicious server attack can be reduced. At last, due to the two-level structure, the security of our framework for MA depends on whether the single-server authentication protocol and the public-key cryptosystem are both secure. So when the security of the single-server authentication protocol and public-key cryptosystem is guaranteed, the MA is secure.

IV. Our Novel Protocol

In this section, a novel protocol instantiated from the generic frameworkforMAwithperfectscalabilityandstronganonymity is elaborated.

A. Proxy Resignature

In our scheme, we introduce the idea of PRS into our design to achieve our goals of perfect scalability and strong user anonymity.ThePRSscheme,whichenablesaproxytoworkasa converteroftwoentities'signatureswasproposedbyBlaze*etal.* [47]. For instance, there exist two clients, Alice (delegatee) and Bob(delegator).TheproxyisabletotransformAlice'ssignature into Bob's signature without their secret keys. Moreover, since the proxy does not own the secret key of either Alice or Bob, it cannot generate a valid signature on behalf of Alice or Bob. In this article, we utilize the PRS scheme proposed by Libert and Vergnaud [48] to achieve this goal. Considering the application scenario of the authentication scheme, the user acts as delegatee and the services provider acts as both delegators and proxy.

B. Setup S_j chooses$_j$ $a_j \in_j Z_q{}^*$, and then computes:

RC chooses two multiplicative cyclic groups (G,G$_T$) with A_j = g_a ,A_{ij} = A_{ai} ,SK $= H_0(\mathrm{ID}_i \| x_i \| A_{ij} \| A_i^{x_j})$, a prime order q and a bilinear map: eˆ : G × G → G$_T$. $M_j = H_0(A_j \| \mathrm{SK})$. Finally, S_j returns message M $\{A\ , M\ \}$ U

Then, RC selects a generator g of G. After that RC selects $_2 =$ $_j$ $_j$ to $_i$. U_i computes: $A_{ij} = A_j^{a_i}$

two secure one-way hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*,\ H_1$: Step 3: $H_0(\mathrm{ID}_i \| x_i \| A_{ij} \| \mathrm{PK}_j^{a_i})$, SK =

{0,1}* → G and a secure symmetric encryption algorithm. Then, U_i verifies S_j

$\mathrm{Enc}_k/\mathrm{Dec}_k(\cdot)$. Finally, RC publishes the system parameters by checking $M_j \overset{?}{=} H_0(A_j \| \mathrm{SK})$. If it holds, U_i $\hat{e}, \mathbb{G}\ \mathbb{G}\ \ , q, g, H\ \ , H$

{ , $_{T\,0\,1}$,Enc$_k$/Dec$_k$(·)}. computes $M_i = H_0(A_i \| \mathrm{SK})$ and sends M_3 = {M_i}

to S_j.

C. Server Registration Phase

Step 4: After receiving the message, S_j confirms U_i has received its message M_2 by checking M_i =$^?$

The SP S chooses x Z^* as its secret key and computes

$_{jj}$ $_j\in$ $_q$ $H_0(A_i \| \mathrm{SK})$

ing (SID$_j$,PK$_j$), RC adds item {SID$_j$,PK$_j$} into list L_S, which is a public database maintained by RC.

other successfully and the session key SK will be used for protecting the following communications between U_i and S_j.

D. User's Registration Phase

In this novel protocol, users do not need to register to the RC. If a user U_i intends to join this system and get services, he/she just need to register to a legal server S_j, which is denoted as the

F. Login and Authentication With the "Foreign" Server The user U_i is able to perform the following steps to interact

PK$_j$ = g^x. Then, S_j submits (SID$_j$,PK$_j$) to RC. After receiv-. If it holds, U_i and S_j authenticate each

U_i's "mother" server. When U_i completes the registration phase, with a server S_k other than his/her "mother" server S_j in this

he/she can access the services provided by his/her "mother" system.

server S_j as well as "foreign" servers in this system. Step 1: U_i inputs the pair of password and iden-

Step 1: U_i selects identity ID$_i$ and his/her password PW$_i$. tity (PW$_i$,ID$_i$). Then, SC$_i$ computes $V_i' =$Then U_i sends ID$_i$ to S_j for registration. $H_0(H_0(\text{ID}_i) \oplus H_0(\text{PW}_i) \bmod n_0)$ and checks

Step 2: After receiving the registration request, S_j checks $V_i' \overset{?}{=} V_i$. SC$_i$ continues to next step only if the whether ID$_i$ is still unused. Then, S_j computes U_is fuzzy verification holds. Then, SC$_i$ recovers U_i's secret key$x_i = H_0(\text{ID}_i\|x_j\|n_i)$ and PK$_i$ = g^{x_i}, the secret key x_i by computing$x_i = N_i \bigoplus H_0(\text{PW}_i)$. resignature key $R_{ij} = \text{PK}_i^{\frac{1}{x_j}}$, where n_i is a ran- $A_i = g^{a_i}$, RID$_i$ chooses $_i = \text{ID}_i \bigoplus H_0(\text{PK}_j^{a_i}), \quad C_{ij} =$ $a_i \in Z_q^*$ and computes:

After that SC

dom number chosen by S_j. And then S_j stores $H_0(\text{ID}_i\|A_i\|x_i\|\text{PK}_j^{a_i}\|\text{SID}_k), \ \sigma_i = H_1(A_i)^{x_i}$. Fi-

(ID$_i$,n_i,R_{ij}) into its database. Finally S_j stores nally, SC$_i$ sends the request {SID$_j$,SID$_k$,A_i,RID$_i$, {SID$_j$,G,G$_T$,q,g,PK$_j$,H_0,H_1,x_i} into a smart C_{ij},σ_i} to S_k.

card SC$_i$ and sends it to U_i securely. Step 2: After receiving this request, S_k forwards it to S_j. Step 3: After receiving the smart card, U_i computes Step 3: Upon receiving the request, S the fuzzy verifier $V_i = H_0(H_0(\text{ID}_i) \oplus H_0(\text{PW}_i)$ RID$_i \bigoplus H_0(A_i^{x_j})$ and then searches$_j$ computesfor nID$_i$ $_i$ of= mod n_0), where n_0 is an appropriate integer U_i from its database. After that, S_j checks $n_0 \in [2^4, 2^8]$ and $N_i = H_0(\text{PW}_i) \bigoplus x_i$. Finally, the $C_{ij} \overset{?}{=} H_0(\text{ID}_i\|A_i\|H_0(\text{ID}_i\|x_j\|n_i)\|A_i^{x_j}\|\text{SID}_k)$

will be omitted from the card.

it holds, S_j chooses $a_j \in Z_q^*$ and searches for the resignature key R_{ij}. Then, S_j computes:

{V_i,n_0,N_i} will be stored into the card and the x_i. If

$$A_j = g^{a_j},\ \sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) = (\sigma_i^{\ j}, \text{PK}_i^{\ j}, R_{ij}^{\ j})^{a} \quad a \quad a$$

a

,

E. Login and Authentication With the "Mother" Server

$V_{ik} = H_0(x_i\|A_i),\ Y_j = \text{PK}_k^{\ j},\ M_{jk} = \text{Enc}_{Y_j}(V_{ik})$.

Step 1: U_i inputs the pair of password and iden- Finally S_j returns message$_k$ {A_j,M_{jk},σ_j,A_i} to S_k.

tity (PW$_i$,ID$_i$). Then, SC$_i$ computes $V_i' =$ Step 4: S_k computes $Y_j = A_j^x$and then uses Y_j to decrypt

$H_0(H_0(\text{ID}_i) \oplus H_0(\text{PW}_i) \bmod n_0)$ and checks ?

$V_i' \overset{?}{=} V_i$. SC$_i$ M continues to the next step only if the $_{jk}$. Then, S_k? checks $\hat{e}(\sigma_{j1},g) = \hat{e}(H_1(A_i),\sigma_{j2})$

fuzzy verification holds. Then, SC$_i$ recovers U_i's and $\hat{e}(\sigma_{j2},g) = \hat{e}(\sigma_{j3},\text{PK}_j)$. If they both hold, S_k secret key x_i by computing $x_i = N_i \bigoplus H_0(\text{PW}_i)$. chooses $a_k \in Z_q^*$, and then computes: $A_k = g^{a_k}$,

After that SC$_i$ $_i$chooses $a_i \in Z_q^*$ and com- $A_{ik} = A_i^{a_k},\ M_k = H_0(\text{SID}_k\|V_{ik}\|A_{ik})$. Finally, S_k putes: $A_i = g^a$, RID $_i = \text{ID}_i \bigoplus H_0(\text{PK}_j^{a_i})$, returns message {$A_k$,$M_k$} to U_i.

$C_i = H_0(\text{ID}_i\|A_i\|x_i\|PK_j^{a_i})$. Finally, SC$_i$ sends Step 5: U_i computes: $A_{ik} = A_k^{a_i}$and then verifies S_k by the request M_1 = {RID$_i$,A_i,C_i} to S_j.checking $M_k \overset{?}{=} H_0(\text{SID}_k\|H_0(x_i\|A_i)\|A_{ik})$. If it

Step 2: Upon receiving the request, S_j computes holds, U_i computes$M_i = H_0(H_0(x_i\|A_i)\|A_{ik})$ and $\text{ID}_i = \text{RID}_i \bigoplus H_0(A_i^{x_j})$ and then searches for n_i sends {M_i} to S_j.

of U_i from its database. After that, S_j checks $C_i \overset{?}{=} H_0(\mathrm{ID}_i \| A_i \| H_0(\mathrm{ID}_i \| x_j \| n_i) \| A_i^{x_j})$. If it holds, U_i and S_j authenticate each other successfully and the session key $\mathrm{SK} = H_0(A_{ik} \| V_{ik})$ will be used for protectingthefollowingcommunicationsbetweenU_i and S_k.

G. Password Updating Phase

The user U_i could locally execute the password updating operation to change his/her password. After completing the fuzzy verification$V_i \overset{?}{=} H_0(H_0(\mathrm{ID}_i) \oplus H_0(\mathrm{PW}_i) \bmod n_0)$,$U_i$ isaccepted to input the new password $\mathrm{PW}_{\mathrm{inew}}$ to generate corresponding new parameters $V_{\mathrm{inew}} = H_0(H_0(\mathrm{ID}_i) \oplus H_0(\mathrm{PW}_{\mathrm{inew}}) \bmod n_0)$ and $N_{\mathrm{inew}} = N_i \oplus H_0(\mathrm{PW}_i) \oplus H_0(\mathrm{PW}_{\mathrm{inew}})$. Finally, V_i and N_i are replaced by V_{inew} and N_{inew} in the user's card.

H. Card Revocation and Reregistration Phase

Once a user U_i find his/her card lost, he/she could perform the following steps to revoke his/her lost card and reregister to the server S_j without the need to change his/her identity.

Step 1: User U_i sends ID_i to S_j with some credentials of him/her.

Step 2: After receiving the reregistration request, S_j first checks the credentials of U_i, e.g., ID-card or passport. Then, S_j computes U_i' s secret key $x_{\mathrm{inew}} = H_0(\mathrm{ID}_i \| x_j \| n_{\mathrm{inew}})$ and $\mathrm{PK}_{\mathrm{inew}} = g^{x_{\mathrm{inew}}}$, the resignature key $R_{\mathrm{ijnew}} = \mathrm{PK}_{\mathrm{inew}}^{\frac{1}{x_j}}$, where n_{inew} is a random number chosen by S_j. And then S_j stores $(\mathrm{ID}_i, n_{\mathrm{inew}}, R_{\mathrm{ijnew}})$intoitsdatabase.Finally,$S_j$ stores $\{\mathrm{SID}_j, \mathsf{G}, \mathsf{G}_T, q, g, \mathrm{PK}_j, H_0, H_1, x_{\mathrm{inew}}\}$ into a smart card SC_i and sends it to U_i.

Step 3: After receiving the smart card, U_i selects a new password $\mathrm{PW}_{\mathrm{inew}}$ and computes the fuzzy verifier $V_i = H_0(H_0(\mathrm{ID}_i) \oplus H_0(\mathrm{PW}_{\mathrm{inew}}) \bmod n_0)$, where n_0 isanappropriateinteger$n_0 \in [2^4, 2^8]$and$N_{\mathrm{inew}} = H_0(\mathrm{PW}_{\mathrm{inew}}) \bigoplus x_{\mathrm{inew}}$. Finally, the $\{V_{\mathrm{inew}}, n_0, N_{\mathrm{inew}}\}$ will be stored into the card and the value of x_{inew} is omitted.

V. RATIONALES OF OUR PROTOCOL

In this part, we will describe some important principles to design the novel protocol: the combination of the following conceptions enable the novel authentication scheme to achieve all security goals and become usable for practice.

A. Two-Factor Authentication

Step 6: After receiving the message, S_k verifies the validity of it by checking$M_i \overset{?}{=} H_0(V_{ik} \| A_{ik})$. If it holds,

The user U_i's secret key $x_i = H_0(\mathrm{ID}_i \| x_j \| n_i)$ is kept in his/her card, where x_j is the serverS_j's secret key and n_i is a random number kept by S_j. For one thing, the U_i's secret authentication key could be computed by the server S_j because it maintains (ID_i, n_i) in its database and preserves x_j secretly. Moreover, the utilization of user's secret key x_i ensures no password verifier table is needed on the server side. For another, x_i is protected by the user's password on the card, so the breach of user's card cannot reveal the user's secret key successfully.

The only way to obtain x_i without the server's key is to obtain the user's two authentication factors simultaneously.

B. Secure Local Verification

Providing secure local verification services is essential in the two-factor authentication mechanism. On the one hand, it could reduce the burden of the server because it could find the invalid input of users locally. On the other hand, it could provide the local password updating service. Thus, a local verifying data should be added into the user's card. However, the introduction of the verifying data may cause additional security risks, such as the smart card loss attack as well as the offline password guessing attack. To solve the problems, in our novel scheme, the local verification data are designed as "fuzzy verifier" [24] : $V_i = H_0(H_0(\mathrm{ID}_i) \oplus H_0(\mathrm{PW}_i) \bmod n_0)$. It is obvious that an adversary may use a wrong pair of (ID,PW) to get the equal value of V_i, which makes A cannot ensure the correctness of his/her guessing password.

C. Strong User Anonymity and Untraceability

Preserving the user's privacy is also crucial as constructing an authentication protocol, which reveals that the user's identity should not be disclosed in the public channel. In our scheme, we use the public-key infrastructure and PRS to protect the user's identityfromdisclosure.First,duringthephaseofauthentication between a user U_i and his "mother" server S_j, the user's real identity ID_I is transformed as $\mathrm{RID}_i = \mathrm{ID}_i \bigoplus H_0(\mathrm{PK}_j^{a_i})$, in which a_i is secretly kept by the server S_j such that the user's identity is effectively protected in the public channel. Second, due to the adoption of PRS [49], the user's privacy can be also protected as the mutual authentication is carried out between this user and a "foreign" server.

D. Authentication in Multiserver Environment

Motivated by the single-server authentication and the technique of PRS, in this article, a novel infrastructure of

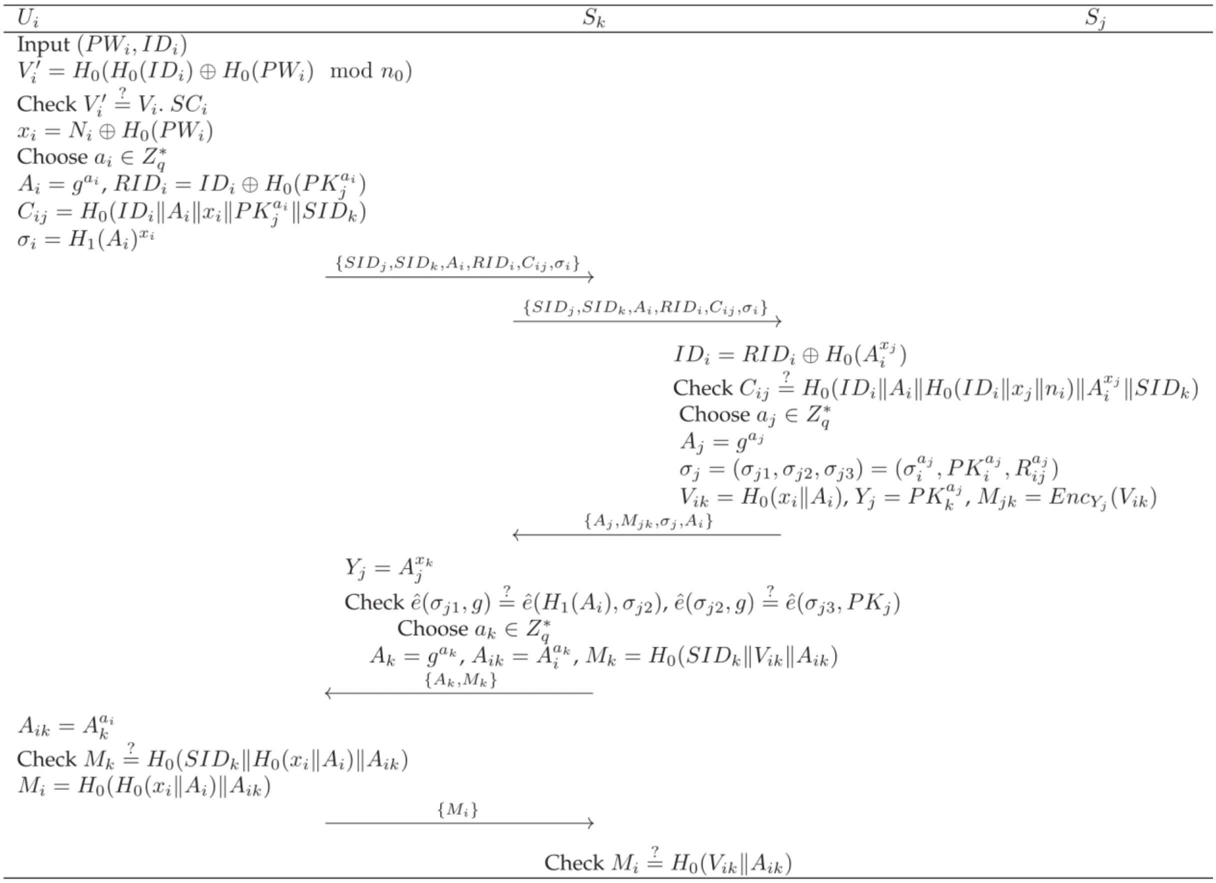

U_i
S_k
S_j
Input (PW_i, ID_i)
$V_i' = H_0(H_0(ID_i) \oplus H_0(PW_i) \mod n_0)$
Check $V_i' \overset{?}{=} V_i$. SC_i
$x_i = N_i \oplus H_0(PW_i)$
Choose $a_i \in Z_q^*$
$A_i = g^{a_i}$, $RID_i = ID_i \oplus H_0(PK_j^{a_i})$
$C_{ij} = H_0(ID_i \| A_i \| x_i \| PK_j^{a_i} \| SID_k)$
$\sigma_i = H_1(A_i)^{x_i}$
$\{SID_j, SID_k, A_i, RID_i, C_{ij}, \sigma_i\}$
$\{SID_j, SID_k, A_i, RID_i, C_{ij}, \sigma_i\}$
$ID_i = RID_i \oplus H_0(A_i^{x_j})$
Check $C_{ij} \overset{?}{=} H_0(ID_i \| A_i \| H_0(ID_i \| x_j \| n_i) \| A_i^{x_j} \| SID_k)$
Choose $a_j \in Z_q^*$
$A_j = g^{a_j}$
$\sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3}) = (\sigma_i^{a_j}, PK_i^{a_j}, R_{ij}^{a_j})$
$V_{ik} = H_0(x_i \| A_i)$, $Y_j = PK_k^{a_j}$, $M_{jk} = Enc_{Y_j}(V_{ik})$
$\{A_j, M_{jk}, \sigma_j, A_i\}$
$Y_j = A_j^{x_k}$
Check $\hat{e}(\sigma_{j1}, g) \overset{?}{=} \hat{e}(H_1(A_i), \sigma_{j2})$, $\hat{e}(\sigma_{j2}, g) \overset{?}{=} \hat{e}(\sigma_{j3}, PK_j)$
Choose $a_k \in Z_q^*$
$A_k = g^{a_k}$, $A_{ik} = A_i^{a_k}$, $M_k = H_0(SID_k \| V_{ik} \| A_{ik})$
$\{A_k, M_k\}$
$A_{ik} = A_k^{a_i}$
Check $M_k \overset{?}{=} H_0(SID_k \| H_0(x_i \| A_i) \| A_{ik})$
$M_i = H_0(H_0(x_i \| A_i) \| A_{ik})$
$\{M_i\}$
Check $M_i \overset{?}{=} H_0(V_{ik} \| A_{ik})$

Fig. 2. Login and Authentication with the “foreign” Server.

authenticationinthemultiserverenvironmentiscontributed.Inthisnovel infrastructure, the RC only needs to act as a CA to authenticate all servers’ public key, which could be done by the government. The user just needs to register to a server (we denote it as the “mother” server of the user) to join this distributed system. The authentication phase between a user and his/her “mother” server is identical to the authentication phase in single-server environment. Besides, the authentication phase between a user and his/her “foreign” server in this system is achieved with the involvement of his “mother” server who relays the information transmitted between the user and the target server by PRS.

In contrast to the traditional authentication scheme for multiserver environment, this novel infrastructure is applicable in real world: the government could act as the CA to maintain and authenticate all servers’ public keys; users could access cooperationservers’servicesofhis/her“mother”serverbyusing the “mother” server’s smart card.

E. Security Requirement

1) Smart card loss attack resistance: As described in *twofactor authentication*, a lost smart card reveals nothing about a user’s secret key in our protocol. Considering this, the presented protocol is secure against the smart card loss attack.

2) Insider attack resistance: In the proposed protocol, it is impossible for an inner adversary to steal the identity and the password from the servers it hacked into. The main reason is that, in our protocol, the servers do not maintain the information about the user’s identity and password.

3) Offline password guess attack resistance: According to *secure local verification*, due to the adoption of the XOR operation, an adversary cannot confirm the value of a user’s password that is stored in the smart card offline. For this reason, our protocol is secure against the offline password guessing attack. 4) Replay attack resistance.

a) Suppose that an adversary A_l has successfully collected the credential (M_1, M_3) generated during the interaction between the user U_i and its “mother” server S_j. Then, A_l sends $M_1 = \{\text{RID}_i, A_i, C_i\}$ to the server S_j who picks a new nonce a_j' and returns $M_2' = \{A_j', M_j'\}$. After receiving this tuple, A_l needs to calculate $A_{ij}' = A_j'^{a_i}$. However, the parameter a_i is pick randomly by the user U_i and is unknown to A_l. Hence, the replay attack fails.

b) On the other hand, if the adversary A_l makes an endeavor to carry out the replay attack to the “foreign” server S_k after snooping the information $\{\text{SID}_j, \text{SID}_k, A_i, \text{RID}_i, C_{ij}, \sigma_i\}$. As described above, the server S_k will forward the information to the user’s

“mother” server S_j who returns $\{A_j', M_{jk}, \sigma_j', A_i\}$.

The server S_k computes $Y_j = A_j'^{x_k}$ to decrypt M_{jk}.

After the verification, S_k returns $\{A_k', M_k'\}$ to $\mathcal{A}_l$ who further compute $A_{ik} = A'^{a_i}_{k}$. The same as the situation mentioned above, the nonce a_i picked randomly by U_i is unknown to A. For this reason, the replay attack to the “foreign” server also fails.

5) Modification attack resistance.

   a) During the login and authentication phase that performed between a user and its “mother” server, the modification executed by an adversary will be discovered because all the interaction messages contains the secret information of either the user or the server.

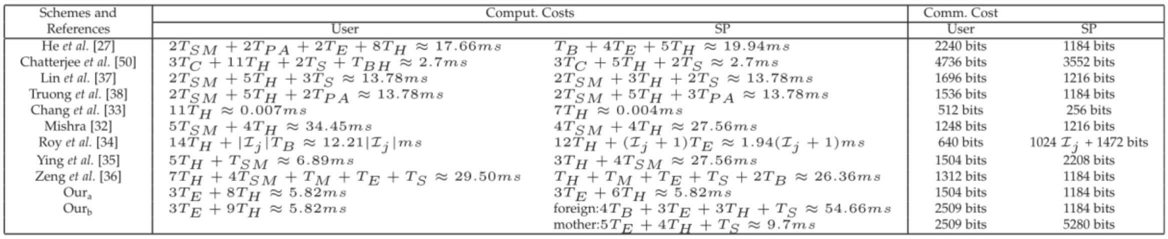
   b) When the user requires services from a “foreign” server, the login and authentication phase is carried out with the involvement of the user’s “mother” server. By introducing the primitive of PRS, the authentication of the massages transmitted between the user and the “foreign” is guaranteed. Hence, the proposed protocol has the ability to resist the modification attack.

6) Server impersonation attack resistance.

   a) *The “foreign” server impersonates the “mother” server:* Once receiving a tuple {RID$_i$,A_i,C_i} from a legitimate user, the adversary needs to recover the real identity of the user as ID = RID$_i$ ⊕ $H_0(A^{x_i j})$, where x_j is the secret key of the user’s “mother” server. Because the adversary is unaware of the “mother” server’s secret key; therefore, the impersonation fails.

   b) *The “foreign” server impersonates the “foreign” server:* In this case, assume that the request from user U_i is originally sent to the server s_k. Once an impersonator S_l receives this message, it normally forwardsittotheuser’s“mother”serverS_j whofinally returns a tuple {A_j,$M_{kj,k}$,σ_j,A_i}. After that, S_l needs to calculate Y_j = A^{x_j} for further decrypting $M_{j,k}$. Nevertheless, the secret key x_k is only known to the server s_k itself. In this way, the impersonation fails.

7) The man-in-the-middle attack resistance: Due to the authentication owned by the proposed protocol, all the modification between users and servers will be detected such that there exists no adversary who can successfully perform main-in-the-middle attack in our protocol.



Schemes and References	Comput. Costs		Comm. Cost	
	User	SP	User	SP
He *et al.* [27]	$2T_{SM} + 2T_{PA} + 2T_E + 8T_H \approx 17.66ms$	$T_B + 4T_E + 5T_H \approx 19.94ms$	2240 bits	1184 bits
Chatterjee *et al.* [50]	$3T_C + 11T_H + 2T_S + T_{BH} \approx 2.7ms$	$3T_C + 5T_H + 2T_S \approx 2.7ms$	4736 bits	3552 bits
Lin *et al.* [37]	$2T_{SM} + 5T_H + 3T_S \approx 13.78ms$	$2T_{SM} + 3T_H + 2T_S \approx 13.78ms$	1696 bits	1216 bits
Truong *et al.* [38]	$2T_{SM} + 5T_H + 2T_{PA} \approx 13.78ms$	$2T_{SM} + 5T_H + 3T_{PA} \approx 13.78ms$	1536 bits	1184 bits
Chang *et al.* [33]	$11T_H \approx 0.007ms$	$7T_H \approx 0.004ms$	512 bits	256 bits
Mishra [32]	$5T_{SM} + 4T_H \approx 34.45ms$	$4T_{SM} + 4T_H \approx 27.56ms$	1248 bits	1216 bits
Roy *et al.* [34]	$14T_H + \lvert\mathcal{I}_j\rvert T_B \approx 12.21\lvert\mathcal{I}_j\rvert ms$	$12T_H + (\mathcal{I}_j + 1)T_E \approx 1.94(\mathcal{I}_j + 1)ms$	640 bits	1024 $\mathcal{I}_j$ + 1472 bits
Ying *et al.* [35]	$5T_H + T_{SM} \approx 6.89ms$	$3T_H + 4T_{SM} \approx 27.56ms$	1504 bits	2208 bits
Zeng *et al.* [36]	$7T_H + 4T_{SM} + T_M + T_E + T_S \approx 29.50ms$	$T_H + T_M + T_E + T_S + 2T_B \approx 26.36ms$	1312 bits	1184 bits
Our$_a$	$3T_E + 8T_H \approx 5.82ms$	$3T_E + 6T_H \approx 5.82ms$	1504 bits	1184 bits
Our$_b$	$3T_E + 9T_H \approx 5.82ms$	foreign:$4T_B + 3T_E + 3T_H + T_S \approx 54.66ms$ mother:$5T_E + 4T_H + T_S \approx 9.7ms$	2509 bits 2509 bits	1184 bits 5280 bits

T_B: Bilinear paring; T_{SM}: scalar multiplication operation of point; T_E: Exponential computation ; T_S: Symmetrical en/decryption computation; T_M:

TABLE I

COMPARISONS OF PERFORMANCE

Map-to-pointmaps;T_{BH}:biohashing;T_H:One-wayhashfunction;T_C:Chebyshevpolynomialcomputation;T_{PA}:PointadditioninECC.Our$_a$:authentication with mother server; Our$_b$: authentication with foreign server. I$_j$: the number of attributes owned by the servers.

$T_B \approx$ 12.21 ms, $T_{\mathrm{SM}} \approx$ 6.89 ms, $T_E \approx$ 1.94 ms, $T_C \approx$ 0.9 ms, T_E,T_M,T_P,T_S,T_{BH},T_H,T_{PA} < 0.01 ms.

VI. Performance Evaluation

In this section, we will compare the performance of the proposed protocol with several state-of-the-art works, i.e., He *et al.* [27], Chatterjee *et al.* [50], Lin *et al.* [37], Truong *et al.* [38], Chang *et al.* [33], Mishra [32], Roy *et al.* [34], Ying *et al.* [35], and Zeng *et al.* [36]. More specifically, we compare our protocol with the symmetric-key MA protocol [33], [37], and the MA protocol [27], [32], [34]–[36], [38], [50] constructed with public-key cryptography systems from the perspective of the computation, communication, and storage cost.

Due to the limited computational ability of the smart card, the computational cost on users side is one of the major concern in assessing the practicality of an authentication protocol. In the comparison, we count the major computations in login and authentication phase, which will be executed much more frequently than the other parts. From the results, it is obvious that our novel protocol involves four exponential computation in multiplicative cyclic groups, denoted by T_E, on the user side. Furthermore, as shown in the bottom of Table I, we measure the computingtimeofthemajorcomputationsbysimulationsonthe common PC (Intel i5-4460, 3.20 GHz, 8 GB memory, Windows XP). From the simulation results, the computation cost in the proposed protocol is a little higher than Chatterjee *et al.*’s protocol and Chang *et al.*’s protocols. However, our protocol is more efficient than He *et al.*’s, Lin *et al.*’s, Truong *et al.*’s, Mishra’s Roy *et al.*’s, Ying *et al.*’s, and Zeng *et al.*’s protocols. Apart from the computational cost, the size of the communication traffic is also important to a practical authentication protocol. In the simulations, we set the size of user identity and the block of symmetric encryption ciphertext as 32 and 128 b, and set the sizeofhashfunctionoutput,thenonceas160bandthegenerator g as 1024 b. In sum, the proposed protocol generates 1504 (with the “mother” server) or 2509 (with the “foreign” server) B communication traffic in one time execution on the users side, which is average. Also, the computation and communication cost on the servers side are very low when a user authenticates with the “mother” server, and the cost is a little higher when a user interacts with the “foreign” server in order to achieve strong anonymity and perfect scalability.

In Table II, we list the security properties summarized in SectionIItoevaluatethestate-of-the-artprotocols.Asillustrated in Table II, our protocol can resist all known attacks and also

can satisfy other security requirements. In contrast, Chatterjee *et al.*'s [50] protocol failed to achieve three-factor security and useranonymity(untraceability)and,thus,wasvulnerabletouser impersonation attack reported in this article. He *et al.* [27]'s self-certified public-key cryptography-based MA protocol was weakagainsttheofflinepasswordguessingattackoncethesmart card was stolen. Lin *et al.* [37] cannot resist offline password guessingattack;TheworkbyTruong*etal.*[38]wassubjecttothe offline password guessing attack and user impersonation attack;
Chang*etal.*[33]wasvulnerabletotheofflinepasswordguessing attack, user impersonation attack, and malicious servers attack; The work by Mishra [32] was under malicious server attack and online password guessing attack; Roy *et al.* confronted the threat of password exposure and failed to resist the smart card loss attack as pointed in [51], and both the works of Ying *et al.* [35] and Zeng *et al.* [36] do not enable two/three-factor security and sound repairability. Hence, both in terms of security as well as practicality, our presented protocol is more feasible.

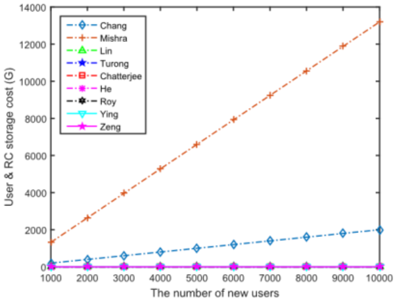


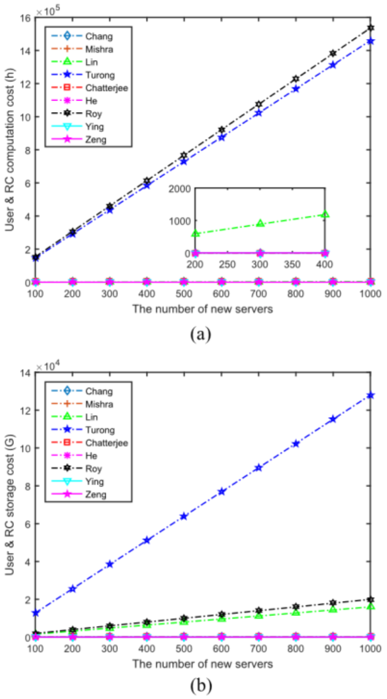
Fig. 4. Scalability on servers side.

Then, in Figs. 3(a), (b), and 4, we compare the scalability of our scheme with that of others and it turns out that our scheme is ofperfectscalabilitywhileothersaresubjecttoanoverwhelming cost. In our simulation experiment, it is supposed that there are 1 000 000 000 users and 10 000 000 SPs in the whole system, and actually the numbers are greater than that in the real world. On the users side, when the system deploys new servers, there exists no extra cost in our scheme. But Lin *et al.*'s protocol [37], Truong *et al.*'s protocol [38], and Roy *et al.*'s protocol are at a distinct disadvantage about this. Concretely, when a new server participates in, it costs users and RC 2.95 hours to

compu
Truong
spend
it will
et al.'s
20 GB
the use
[37] as
protoco
progres
the dat
that wi
collaps
system
200 GB
protoco
possibl
is no ex
that ou
users a
Ther
scalabi
for tha

TABLE II COMPARISONS OF SECURITY PROPERTIES

Schemes and References	C1	C2	C3	C4	C5	C6	C7
He *et al.* [27]	√	√	√	√	√	√	√
Chatterjee *et al.* [50]	√	√	√	√	×	√	√
Lin *et al.* [37]	×	√	√	√	√	√	×
Truong *et al.* [38]	√	√	√	√	√	×	√
Chang *et al.* [33]	√	√	√	√	√	×	√
Mishra [32]	√	√	√	√	√	√	×
Roy *et al.* [34]	√	√	√	√	√	√	√
Ying *et al.* [35]	√	√	√	×	√	×	×
Zeng *et al.* [36]	√	√	√	×	×	×	√
Our	√	√	√	√	√	√	√



(a)

(b)

Fig. 3. Scalability on users side. (a) Computation cost. (b) Storage cost.

satisfies the security properties as well as perfect scalability. In sum, the above-mentioned comparisons indicate that not only a novel two-factor MA protocol is proposed but also a better tradeoff between efficiency, scalability, and security is also achieved.

VII. Conclusion

In this article, we analyzed the goal of the scalability for MA protocol. With the consciousness of modular design, we constructed a generic framework from single-server authentication to MA. Giving an instantiation of the framework, we proposed a novel MA protocol that satisfied the 14 security requirements by combining two-factor single-server authentication and PRS. Importantly, in the comparison with six protocols, we proved that our novel protocol is of perfect scalability and strong user anonymity without losing efficiency on the users side, which means that the novel protocol is well suited for either practical application and high-security applications.

References

[1] "Number of smartphone users worldwide from 2014 to 2020 (in billions)," 2016. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-w orldwide/

[2] E. Turban, J. Outland, D. King, J. K. Lee, T.-P. Liang, and D. C. Turban, “Mobile commerce and the internet of things,” in *Proc. Electron. Commerce*, 2018, pp. 205–248.

[3] D. Korzun, “Internet of Things meets mobile health systems in smart spaces: An overview,” in *Internet of Things and Big Data Technologies for Next Generation Healthcare*. New York, NY, USA: Springer, 2017, pp. 111–129.

[4] M. Backes, M. Gagné, and M. Skoruppa, “Using mobile device communication to strengthen e-voting protocols,” in *Proc. 12th ACM Workshop Workshop Privacy Electron. Soc.*, 2013, pp. 237–242.

[5] R. Žitny`, T. Szabó, I. Pšenáková, Z. Illés, and V. Bakonyi, “Using mobile technologiesinuniversityeducation,”in*Proc.Int.Conf.Emerg.eLearning Technol. Appl.*, 2016, pp. 387–392.

[6] L. Lamport, “Password authentication with insecure communication,” *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[7] “state-sponsored yahoo hack exposed 500 m users,” Sep. 2016. [Online]. Available: http://nypost.com/2016/09/22/yahoo-hackers-breachedat-least-500m-accou nts-in-2014/

[8] Y.Xue,“ChineseInternetsuffersthemostserioususerdataleakinhistory,” Dec. 2011. [Online]. Available: https://blogs.forcepoint.com/securitylabs/chinese-internet-suffers-mos t-serious-user-data-leak-history

[9] T. Pham, “Four years later, anthem breached again: Hackers stole credentials,” Feb. 2015. [Online]. Available: https://duo.com/blog/four-yearslater-anthem-breached-again-hackers-stole-em ployee-credentials

[10] D. Wang, D. He, P. Wang, and C.-H. Chu, “Anonymous two-factor authenticationindistributedsystems:Certaingoalsarebeyondattainment,”*IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.

[11] A. Juels and M. Wattenberg, “A fuzzy commitment scheme,” in *Proc. 6th ACM Conf. Comput. Commun. Secur.*, 1999, pp. 28–36.

[12] J. Lee, S. Ryu, and K. Yoo, “Fingerprint-based remote user authentication scheme using smart cards,” *Electron. Lett.*, vol. 38, no. 12, pp. 554–555, 2002.

[13] H.-S. Kim, S.-W. Lee, and K.-Y. Yoo, “Id-based password authentication scheme using smart cards and fingerprints,” *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 4, pp. 32–41, 2003.

[14] A. Bhargav-Spantzel, A. C. Squicciarini, S. Modi, M. Young, E. Bertino, and S. J. Elliott, “Privacy preserving multi-factor authentication with biometrics,” *J. Comput. Secur.*, vol. 15, no. 5, pp. 529–560, 2007.

[15] C.-I. Fan and Y.-H. Lin, “Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics,” *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 4, pp. 933–945, Dec. 2009.

[16] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, “A generic framework for three-factor authentication: Preserving security and privacy in distributed systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, Aug. 2011.

[17] D. He and D. Wang, “Robust biometrics-based authentication scheme for multiserver environment,” *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.

[18] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, “Robust extended chaoticmaps-basedthree-factorauthenticationschemepreservingbiometric template privacy,” *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2085–2101, 2016.

[19] L.-H. Li, L.-C. Lin, and M.-S. Hwang, “A remote password authentication scheme for multiserver architecture using neural networks,” *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1498–1504, Nov. 2001.

[20] W.-S. Juang, “Efficient multi-server password authenticated key agreement using smart cards,” *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 251–255, Feb. 2004.

[21] V. Odelu, A. K. Das, and A. Goswami, “A secure biometrics-based multi-server authentication protocol using smart cards,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[22] Y.-M. Tseng, S.-S. Huang, T.-T. Tsai, and J.-H. Ke, “List-free id-based mutual authentication and key agreement protocol for multiserver architectures,” *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 102–112, Jan.–Mar. 2016.

[23] D. Wang and P. Wang, “On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions,” *Comput. Netw.*, vol. 73, pp. 41–57, 2014.

[24] D. Wang and P. Wang, “Two birds with one stone: Two-factor authentication with security beyond conventional bound,” *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2016.

[25] C.-M. Hsiao and Y.-P. Liao, “A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients,” *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 886–900, Mar. 2013.

[26] M.-C.ChuangandM.C.Chen,“Ananonymousmulti-serverauthenticated key agreement scheme based on trust computing using smart cards and biometrics,” *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1411–1418, 2014.

[27] D. He, S. Zeadally, N. Kumar, and W. Wu, “Efficient and anonymous mobileuserauthenticationprotocolusingself-certifiedpublickeycryptography for multi-server architectures,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2052–2064, Sep. 2016.

[28] A. K. Das, S. Jangirala, and S Mukhopadhyay, “A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards,” *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2735–C2767, 2017.

[29] R. S. Pippal, C. Jaidhar, and S. Tapaswi, “Robust smart card authentication scheme for multi-server architecture,” *Wireless Pers. Commun.*, vol. 72, no. 1, pp. 729–745, 2013.

[30] W. Liu, X. Hu, and W. Jianghong, “Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture,” *Wireless Pers. Commun.*, vol. 77, no. 3, pp. 2255–2269, 2014.

[31] K.-H. Yeh, “A provably secure multi-server based authentication scheme,” *Wireless Pers. Commun.*, vol. 79, no. 3, pp. 1621–1634, 2014.

[32] D. Mishra, “Design and analysis of a provably secure multi-server authentication scheme,” *Wireless Pers. Commun.*, vol. 86, no. 3, pp. 1095–1119, 2016.

[33] C.-C. Chang, T.-F. Cheng, and W.-Y. Hsueh, “A robust and efficient dynamic identity-based multi-server authentication scheme using smart cards,” *Int. J. Commun. Syst.*, vol. 29, no. 2, pp. 290–306, 2016.

[34] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. Rodrigues, “Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 457–468, Jan. 2019.

[35] B. Ying and A. Nayak, “Lightweight remote user authentication protocol formulti-server5Gnetworksusingself-certifiedpublickeycryptography,” *J. Netw. Comput. Appl.*, vol. 131, pp. 66–74, 2019.

[36] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, “E-AUA: An efficient anonymous user authentication protocol for mobile IoT,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, 2018.

[37] F. Wen, C. Du, and H. Lin, “An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics,” *Wireless Pers. Commun.*, vol. 84, no. 4, pp. 2351–2362, 2015.

[38] T.-T. Truong, M.-T. Tran, A.-D. Duong, and I. Echizen, “Provable identity based user authentication scheme on ECC in multi-server environment,” *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2785–2801, 2017.

[39] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo, “Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment,” *IEEE Access*, vol. 5, no. PP, pp. 3622–3639, 2017.

[40] X. Li, J. Niu, S. Kumari, J. Liao, and W. Liang, “An enhancement of a smart card authentication scheme for multi-server architecture,” *Wireless Pers. Commun.*, vol. 80, no. 1, pp. 175–192, 2015.

[41] B.WangandM.Ma,“Asmartcardbasedefficientandsecuredmulti-server authentication scheme,” *Wireless Pers. Commun.*, vol. 68, pp. 361–378, 2013.

[42] D. Mishra, A. K. Das, and S. Mukhopadhyay, “A secure user anonymitypreserving biometric-based multi-server authenticated key agreement scheme using smart cards,” *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8129– 8143, 2014.

[43] C. Wang, X. Zhang, and Z. Zheng, “Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme,” *Plos One*, vol. 11, no.2, 2016, Art. no. e0149173.

[44] S. A. Chaudhry, “A secure biometric based multi-server authentication scheme for social multimedia networks,” *Multimedia Tools Appl.*, vol. 75, no. 20, pp. 12 705–12 725, 2016.

[45] S. Kumari *et al.*, “A provably secure biometrics-based authenticated key agreement scheme for multi-server environments,” *Multimedia Tools Appl.*, vol. 77, pp. 2359–2389, 2018.

[46] J. Yu, G. Wang, Y. Mu, and W. Gao, “An efficient generic framework for three-factor authentication with provably secure instantiation,” *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2302–2313, Dec. 2014.

[47] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxycryptography,”in*Proc.Int.Conf.TheoryAppl.Cryptograph.Techn.*, 1998, pp. 127–144.

[48] B.LibertandD.Vergnaud,“Multi-useunidirectionalproxyre-signatures,” in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 511–520.

[49] G. Ateniese and S. Hohenberger, “Proxy re-signatures: New definitions, algorithms,andapplications,”in*Proc.12thACMConf.Comput.Commun. Secur.*, 2005, pp. 310–319.

[50] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, “Secure biometric-based authentication scheme using Chebyshevchaoticmapformulti-serverenvironment,”*IEEETrans.Depend.Sec. Comput.*, vol. 15, no. 5, pp. 824–839, Oct. 2018.

[51] D. Wang, X. Zhang, Z. Zhang, and P. Wang, “Understanding security failures of multi-factor authentication schemes for multi-server environments,” *Comput. Secur.*, vol. 88, 2020, Art. no. 101619.

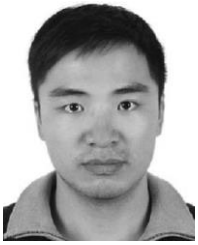
Hu Xiong received the Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009.

He is currently a Full Professor with the UESTC. His research interests include public key cryptography and networks security.

Zhiqing Kang received the B.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019. She is currently a Software Engineer with Tencent, Shenzhen, China. Her research interests include public key cryptography and network security.

Jinhao Chen received the B.S. degree from the Sichuan University of Science and Engineering, Zigong, China, in 2018. He is currently working toward the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China.

Hisresearchinterestsincludeforwardsecuritypublic key cryptography.



Junyi Tao received the B.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2017 and the M.S. degree from the University of Southern California, Los Angeles, CA, USA, in 2019. He is currently workingtowardthePh.D.degreewiththeDepartment of Computer Science and Engineering, Stony Brook University, Stony Brook, NY, USA.

He is also affiliated with the National Security Institute, Stony Brook University. His research interests include applied cryptography and usable security.

ChenYuanreceivedtheM.S.degreefromtheSchool of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2017. He is currently working towardthePh.D.degreewiththeDepartmentofComputer Science and Engineering, University at Buffalo, Buffalo, NY, USA.

His research interests include secure multiparty computation and authentication.

Saru Kumari received the Ph.D. degree in mathematics from Chaudhary Charan Singh University, Meerut, India, in 2012.

She has authored and coauthored more than 149 research papers in reputed international journals and conferences, including 131 publications in SCIindexedjournals.Sheisontheeditorialboardofmore than a dozen international journals of high repute under Elsevier, Springer, Wiley, and others including SCI journals. Her current research interests include cryptology, information security, digital authentication, and security of wireless sensor networks.