Smartphone based User Authentication Scheme without Verifier Table on Authentication Server

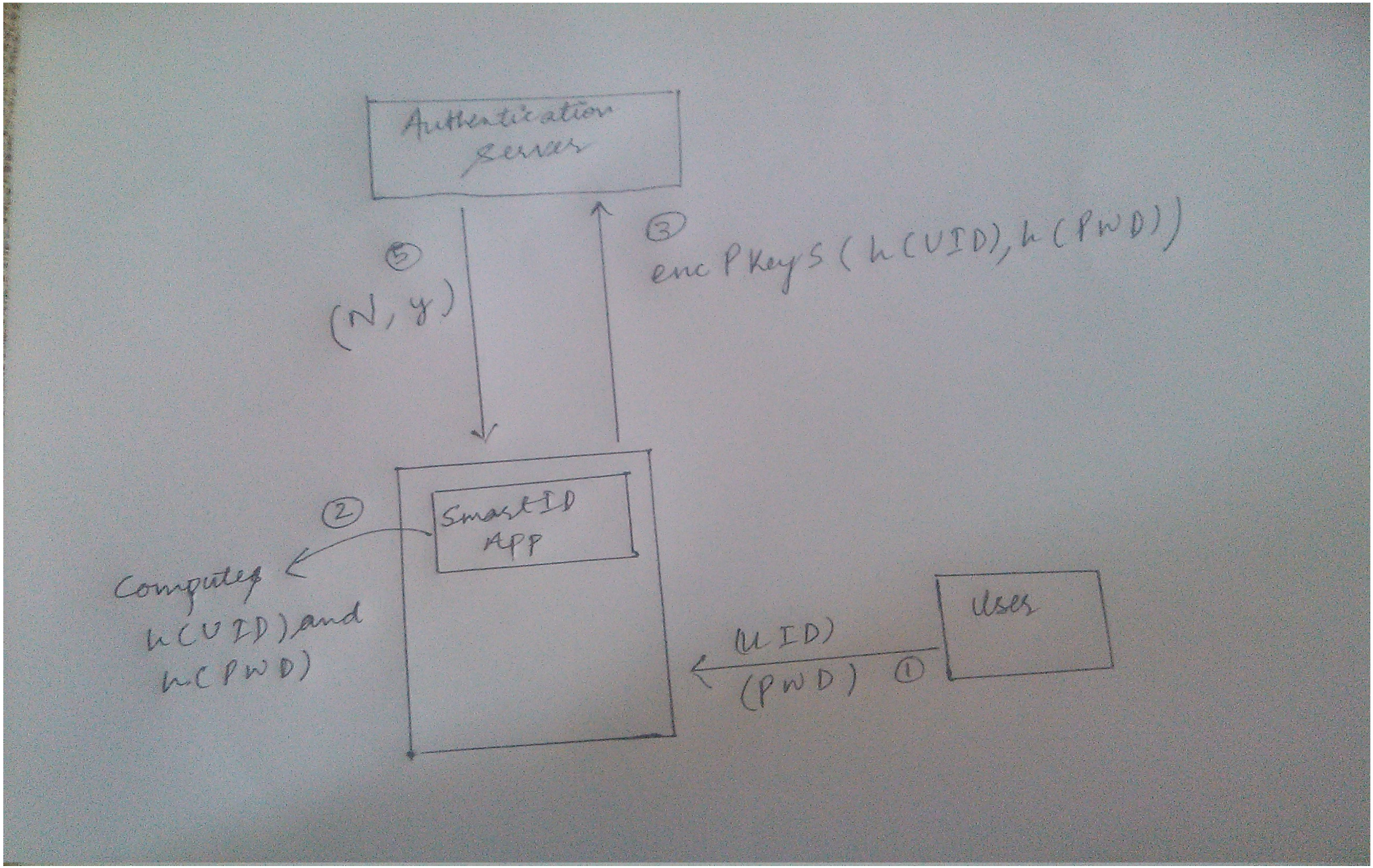
Bhaskar Kalia
Anurag Singh
Sushant Thakur
Premlata Negi
Nisha Kumari

The Proposed Scheme

The proposed scheme also consists of three phases

- Registration phase: Used for registration of new user
- Authentication phase: Used for every time user want to access resource
- Password change phase: Used when user loses his phone to recover the account

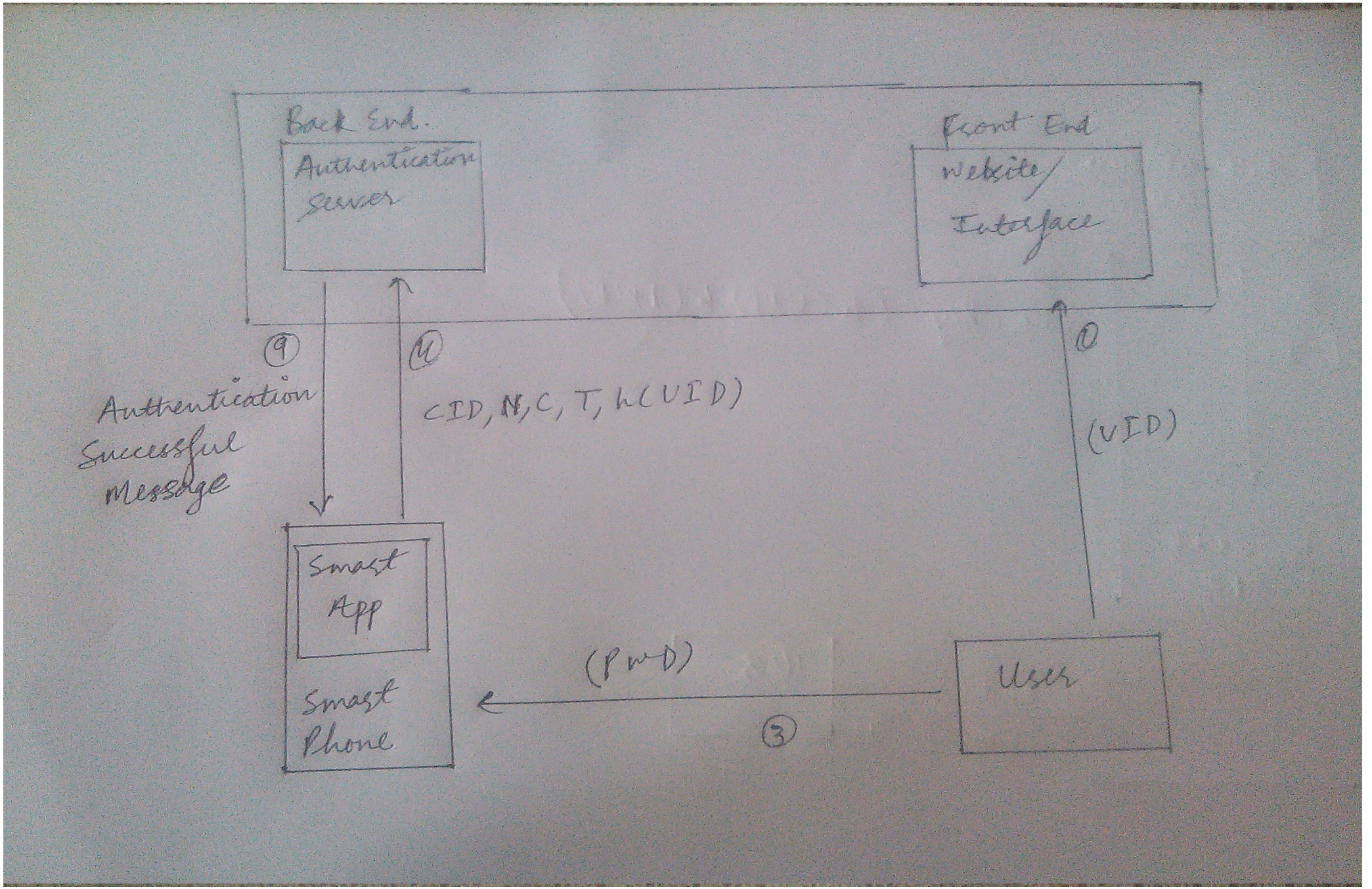
Registration Phase



1. User will start the Application on phone and tap on the "Register" option to register himself. A login page will display in which user will enter the UserID (**UID**), password(**PWD**).
2. Application computes **h(UID)** and **h(PWD)**.
3. Application encrypts **h(UID)** and **h(PWD)** with the **Public Key** of Authentication server and sends to Authentication server.

4. Server computes Nonce,
   $N = h(PWD) \oplus h(x)$, where **x** is server master key.
5. Server sends back **N** and **y** to application where **y** is a number stored in each user application.

Authentication Phase



1. User enters **UID** on website/server .
2. Website directs user to authenticate via app.
3. User enters **PWD** in the application and taps on authenticate button.
4. Application computes,
   CID= h(PWD) ⊕ h(N ⊕ y ⊕ T)*,*
   *where **T** is the curent date and time.*

   B= h(CID ⊕ h(PWD)). -
   C= h(T ⊕ *N* ⊕ *B* ⊕ *y)*.

   App->Server: CID,N,C,T,h(UID).

5. Upon receiving the login message(***CID,N,C,T, UID***) at the time ***T**** , the remote server authenticates the user with the following steps:

6. Verify the validity of the time interval between ***T*** and ***T**** . If **(*T**-*T*) >=*del* *T*** , where *del* *T* denotes the expected valid time interval for transmission delay, then the remote server rejects the login request.

7. Computes ***h(PWD)= CID ⊕ h(N⊕ y⊕ T)*** .

8. Computes ***B= h(CID⊕ h(PWD))***.

9. Thereafter, checks whether $C = h(T \oplus N \oplus B \oplus y)$. If it holds,the remote system accepts the login request. Otherwise it terminates the login operation .

Password Change Phase

This phase is invoked whenever the user wants to change his password. He can easily change his password without taking any assistance from the remote system. The phase works as follows:

1. The user opens his application. He submits the password ***PWD*** and requests to change the password.
2. Then *User* chooses new password ***PWD**** .

3. The smart card computes

$N^* = N \oplus h(PWD) \oplus h(PWD^*)$, which yields $h(PWD^*) \oplus h(x)$.

4. The nonce N will be replaced with N^*. The password has been changed with the new password PWD^* and terminates the operation.