Authentication Scheme for mobile cloud Computing Services

Abstract

Because the authentication method based on username-password has the disadvantage of easy disclosure and low reliability and the excess password management degrades the user experience tremendously, the user interested to get rid of the bond of the password in order to seek a new way of authentication. Therefore, the multifactor biometrics-based user authentication wins the favor of people with advantages of simplicity, convenience, and high reliability. Now the biometrics-based (especially the fingerprint information) authentication technology has been extremely mature, and it is universally applied in the scenario of the mobile payment. Unfortunately, in the existing scheme, biometric information is stored on the server side. As thus, once the server is hacked by attackers to cause the leakage of the fingerprint information, it will take a deadly threat to the user privacy. Aiming to solve the security problem of the fingerprint information in the mobile network environment we proposed an authentication scheme based on three factors and implement hash, XOR operations and fuzzy extractor algorithm that extract a key from fingerprint information without exposing that information to the server's attackers.

Hardware parts

- Smart Mobile Device (SMD) → Samsung Galaxy Note 5.
- External fingerprint scanner (EFS) device → Digital Persona (U.are.U 4500) as a fingerprint scanner with USB 2.0 cable

Software parts

- Integrated development environment to build android application →Android Studio 3.5.3.
- Programing languages → Java, Python and Node JS.
- Web server → XAMPP (Apache and MySQL).

Registration Phase

During this phase, our work depends on two steps: the first is related to Admi that manages the system while the second connects with AS.

StepR1: User Side: User should registered into AS, they depend on their SMD that contain registration application to complete this authentication process. This step can described in the following points:

- A user first freely selects a 4-digts PIN, enters it twice into the app to ensure that the PIN have entered correctly.
- After the conforming of the PIN mobile app ask the user to enroll his/her finger into the external fingerprint scanner ,process the fingerprint image and send it to fizzy extraction algorithm in order to generate R and P
  $(R,P)=Gen(FP)$.
- While fingerprint enrollment and processing, the mobile app extract the unique identity of the mobile device (DID)
- The mobile app compute KF, KP and KI where:

$$KF = h(R ,PIN) \qquad KP = h(R,DID) \qquad KI=h(KF \oplus KP)$$

- After that, the mobile app create a Credential file (CFSMD) to save the important parameters (p) into SMD secretly.
- Mobile app send a message M that contains important parameters:

  M= (KI) to AS

StepR2: Authentication Server Side:

Upon receiving, a message from SMD, AS implements the following steps:

- checks the Index File in the database and compares KI` (receiving) ? = KI (existing) in server data base; if the result is equal, Admi has been registered in the system. Otherwise, they create a new record to save KI of the user.

Login Phase:

StepL1: User Side:

When admin wishes to log into the system, they should perform the following steps:

- They enter PIN and Fingerprint (FP) by using a mobile device (SMDi) to allow them to use the important parameters inside CFSMD.
- SMD applies the reproduction function of the fuzzy extractor to calculate R` = Rep (P,FP).
- SMD send message M`= (KI) to AS

StepL2: Authentication server side:

- AS checks KI (receiving) ? = KI (existing) in IFAS if the resut is equled ,User gain access to his/her account in the AS.If not the access will denied .

Work that have been done

1.registration stage in wich the PIN and mobile device number are obtained and sent to an external database (MySQL) by means of Node Js .

2.Verification stage (PIN and mobile device number) .Both points 1 and 2 are for teasting only untill now .

Work that haven’t been done

1.Implement fuzzy extractor algorithm into the fingerprint data to produce R (key) and P (helper data) .

2.kept P in app. Internal database and send hashed value of R to the external database (mysql)