Challenge Token-based Authentication – CTA

Ghassan Kbar
Associate Research Professor
Riyadh Techno Valley, King Saud University
Riyadh, Saudi Arabia
gkbar@ksu.edu.sa, gahkbar@yahoo.com

***Abstract*- A new technique for highly securing the wired and wireless local area network using a Challenge Token-based Authentication as a second authentication factor is presented. This technique is based on two authentication factors, which is in addition to the first factor "user name and password", it also uses the client soft token that will be stored in a mobile phone or USB. The soft token will be obtained during registration and will never be transmitted during the authentication process. This token will be used by a mobile Client Program to generate a secure Authentication Server (AS) public key in order to respond to the AS's challenge. This new authentication mechanism addresses the vulnerabilities existed in the existing weak authentication method that is based only on first authentication factor. It would also solve the Denial of Service attack existed in the second authentication factor techniques because a secure server public key is used instead of well known server public key such as the one used in Extensible Authentication Protocol, and Wireless Application Protocol. In addition it reduces the complexity and associated cost existed in the mobile phone authentication technique since there is no need to send sms messages to authenticate the clients. In addition, the session key will be exchanged using the derived secure AS public key that is correlated to the soft token. This makes the security parameters known only to authentication server, and valid supplicants. Attackers would be unlikely able to know the token and other security keys since the token is only exchanged during registration through a trusted party. Moreover, the use of 2 authentication factors would make the security system stronger and more relevant to sensitive applications particularly for banks.**

Keywords-wireless; authentication; security.

I. INTRODUCTION

Strong authentication is extremely needed for e-transaction applications to secure the credential information including the one used in credit card. User name and password are widely used as the main authentication mechanism in computer security systems including the e-transaction applications. This mechanism has various drawbacks including the poor selection of password, the vulnerability to capture and crack password especially in wireless Local Area Network (LAN) and Wireless LAN (WLAN). Hence, the use of one factor authentication that is known as "something you know", and is mainly used in today system security, would not be sufficient for important secure applications.

The issue of authentication becomes more vulnerable in WLANs based on 802.11 standards if no authentication or weak encryption key is used. Although attempts are continuously being made to address the security issues in the later versions of 802.11 [1], security of WLAN remains challenging. Despite the enhancements provided by Wired Equivalent Privacy (WEP) for WLANs, the demands for a further secured environment still a high priority issues in wireless network. By using the Extendable Authentication Protocol (EAP) for authentication, the access point responds by enabling a port for passing only the EAP packets from the client to an authentication server located on the wired side of the access point. Wi-Fi Protected Access (WPA) [2] also authenticates securely the wireless users to the network. WPA is a subset of the abilities of 802.11i, including better encryption with Temporal Key Integrity Protocol (TKIP), easier setup using a pre-shared key, and the ability to use RADIUS-based 802.1X authentication of users [3]. Wireless security will continue to be a concern for the foreseeable future. Although a single overall solution has yet to be perfected, the best protection is always prevention [4]. Comparison between the different EAP products has been presented in [5]. The issues of flexibility and high speed authentication technique, has been addressed using token-based fast Authentication method for Wireless network [6].

To strengthen the authentication system and improve the system security further, at least two modes of authentication should be used [7]. Hardware token is used in some applications to increase the security during authentication. However, the hardware token has its own limitations such as inconsistent availability, and loose of token sometimes. The primary mode of authentication would rely on user name and password, while the second authentication can be used for emergency in case the primary one is unavailable to users. There are many methods used for emergency authentications [8], but it will still be considered as one way authentication since it would only be used in case of forgetting the user name and password. Passwords and emergency type of life questions are often considered as “something you know,” while hardware tokens are considered as “something you have.” Another category of authenticator is “something you are,” such as the biometric authentication. Some research try to introduce a forth-factor authentication "Somebody you know" [8], but still be considered as the first method by allowing other people to verify the users in case of emergency. A software token can be used as a second method of authentication and would be considered as "something you have". This combination of authentications is easy to implement and can achieve the purpose of having a strong authentication mechanisms.

 ISBN: 978-1-61208-140-3

Unlike the combination of the first method with hardware token which has its own limitation, or the combination with biometric authentication which is difficult to use in e-transaction applications, the combination with Soft token is preferable for ease of implementation as described in section IV. However, the transmission of the soft token has to be secure. Using tokens involves several steps including registration of users, token production and transmission, user and token authentication, and token revocation [10]. Some bank applications use a second authentication factor by sending an SMS message with code to registered mobiles, where the user has to reenter this code in order to continue the authentication process [11, 12]. This mechanism is costly, relies on technology such as GSM which might not be available all the time, and suffers from man-in-the-middle MiM attacks. Encrypting the SMS message would solve the MiM attacks as presented in [13]. Since many people carry a mobile phone at all times, an alternative is to install all the software tokens on the mobile phone [14]. This proposed system involves using a mobile phone as a software token for One Time Password (OTP) generation. However, if the mobile is out of sync with the server, it has to send a SMS message requesting the server to generate OTP and send it back to the mobile. This system might not be reliable and would be costly if it was out of sync where at least 2 SMS will be send to obtain the OTP.

In this paper a new proposed method for transmitting the soft token is used securely, where the token is only transmitted during registration through a trusted Registry Authority (RA) or trusted Authenticators in WLAN, and will be stored in a mobile phone or USB along with a secure key algorithm. The supplicant will *respond to a challenge request from the AS and forward it to the mobile or USB, where it will use the stored soft token to derive a secure server public key to communicate with the AS in order to securely exchange the session key* as will be described in the next section. The use of soft token in this paper is not to generate OTP or to send SMS message with code, but in fact to generate a secure AS public key that is correlated to the temporary key sent by the server, the stored token, and the secure key algorithm which all known only to valid supplicants and server. This will prevent the Denial of Service (DoS) attacks existed in other techniques, where attackers cannot use the well known server public key during authentication and are unable to generate the secure server public key which require a valid token and secure key Algorithm.

II. CHALLENGE TOKEN BASED AUTHENTICATION (CTA) CONCEPT

In order to have a strong authentication mechanism when users want to access sensitive information from an Application Server (APS), 2 authentication factors should be used. Using user name and password (something you know) as the first authentication factor, in addition to the challenge/response token based authentication (something you have) will strengthen the authentication mechanism. Where the soft token will be issued by the AS to supplicants during the registration, and later can be used by the supplicant to generate a secure AS public key (SsP) by applying a certain algorithm known to AS and to valid supplicants only. Only supplicants that can respond to the AS's challenge, by generating the SsP, would be able to communicate to AS in order to obtain the session key (SK) which will be used to communicate with the application server APS. In order to implement the concept of the challenge token based authentication (CTA), there is a need for a third party authority such as the Registry Authority (RA) server which is responsible for issuing a Temporary Registry Token (RT) for the valid supplicants. These supplicants will then use the RT to register with the AS and obtain a Permanent Token (PT) and a Secure Key Algorithm (SKA). The RT and SKA will be used by the supplicant to generate an SsP in order to communicate to the AS and to respond to the challenge request.

Figure 1 describes the different parameters that will be created and used by each party involved in the authentication process. At the user/client terminal, there are different authentication parameters used for registration/authentication, where the client terminal passes the username, password, client name and the client registry token (RT) to the RA, and to the Authentication Server (AS). The RA will then respond by passing a Permanent Token (PT) and a Secure Key Algorithm (SKA), which will be stored at the client side for future authentication. In addition, there are different encryption parameters used during registration, authentication, and transfer of messages. The public/private keys of client, registry authority and server are used to encrypt messages during the registration phase. While the secure public/private keys that are generated by the client terminal and server, are used to encrypt messages during the authentication phase. A symmetric session key is used for encrypting the messages sent between the client terminals and application servers.

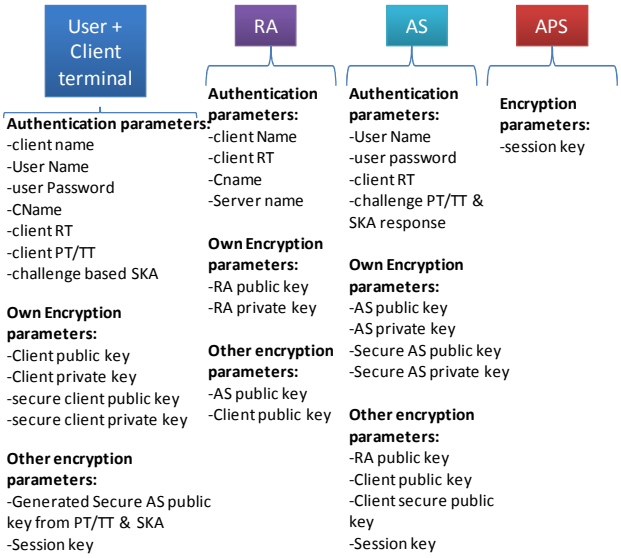

Figure 1. The different encryption parameters used by all parties during authentication

Figure 2 describes the steps required for registration with RA and AS, as well as the steps that are used during

authentication with the AS and during transmission of messages to the APS. The following steps describe the different phases of registration, authentication and access of applications:

- **AS Server to Register Authority (RA) Registration:**
  - (step 1,2): AS uses **VPN** to register to RA using RA's public/private key.
- **Client to Register Authority RA authentication:**
  - (step 3) Client registers with RA by passing user credential (user name UN, and client public key cP) encrypted by RA's public key.
  - (step 4) RA passes the RT to supplicant which will be used by supplicant to continue registration with the AS.
  - (step 5) Supplicant passes Challenge Name **CName** to be stored in RA for future retrieval of registry token.
  - (step 6) Supplicant requests the public key of AS by passing UN and AS flag encrypted by RA's public key to register with AS for future communication.
  - (step 7) RA passes the AS's public key to supplicant encrypted by client public key cP.
- **Supplicant to AS Registration:**
  - (step 8) Supplicant registers to AS by passing the RT and its public key cP encrypted by AS's public key sP.
  - (step 9) AS request client registry validation to RA using VPN.
  - (step 10) RA validate supplicant by checking its RT and reply to AS
  - (step 11) Following the supplicant validation, AS will send the PT and the SKA to the supplicant.
  - (step 12) The supplicant will forward the PT and SKA to mobile using Bluetooth technology or to USB which will then be stored in order to be used later to generate a secure server public key (SsP). The SsP will be used by supplicant to respond to challenge received from AS in order to complete the authentication between the supplicant and the AS.
  - (step 13) Supplicant will pass the user name UN and password PW encrypted by SsP to AS in order to be stored at the AS for future accessing the application servers. The AS will store UN and PW for further authentication process.
  - (step 14) AS sends registration complete the supplicant
- **Client Authentication to AS:**
  - (step 15) supplicant request access to the application server APS through AS, by sending its UN and PW encrypted by server public key sP.
  - (step 16) AS sends a challenge authentication request to the supplicant by passing a temporary server public key TsP encrypted by the supplicant public key cP.
  - (step 17) supplicant will decrypt the challenge request, extract the TsP and forward it to the mobile or USB in order to get the secure AS public key SsP.
  - (step 18) the mobile or USB, will receive the request from the supplicant and if user accepts it by pressing OK, its client token program will generate the SsP from the stored PT and received TsP by using the stored SKA algorithm and pass it to supplicant.
  - (step 19) supplicant will generate a secure client public key ScP, and send the ScP encrypted by SsP to the AS.
  - (step 20) AS decrypts the challenged message to obtain the ScP, and use it to encrypt and send a session key SK to the supplicant.
  - (step 21) the supplicant will send the SK to the application server APS.
- **Exchange of messages between supplicant and APS**:
  - (step 22) the supplicant sends a message m1 to the APS encrypted by the SK.
  - (step 23) the APS will decrypt the message m1 using the SK and send message m2 encrypted by SK to the supplicant. The supplicant will be able to decrypt the message m2 using the same session key SK.

Advantages of the CTA mechanism:

- 2 authentication mechanisms are used; user name, password used as first authentication factor and client token for server challenge as second authentication factor.
- Token is never transferred during authentication, but is only transferred during registration to the RA and the AS, or it can be obtained through hardware USB, or it can be retrieved from the RA by passing challenge name CName.
- Used a derived secure AS public key SsP that is based on a stored token and correlation SKA algorithm to respond to AS challenge. This derived SsP can only be derived by the supplicant which it has the valid token and therefore hackers would not be able to communicate with the AS since they cannot generate the SsP.
- Exchange the session key SK using SsP is only known to valid supplicants. This will prevent attackers from using the normal server public key sP to launch a DOS attack at the AS.
- Compared to other second authentication factor used in other methods such as the one using hard or soft token for generating One Time Password or sending SMS message, the complexity in CTA is less, the cost is cheaper, and moreover the DOS attacks are eliminated since a secure server public key is used which is only known to valid supplicants.

 ISBN: 978-1-61208-140-3

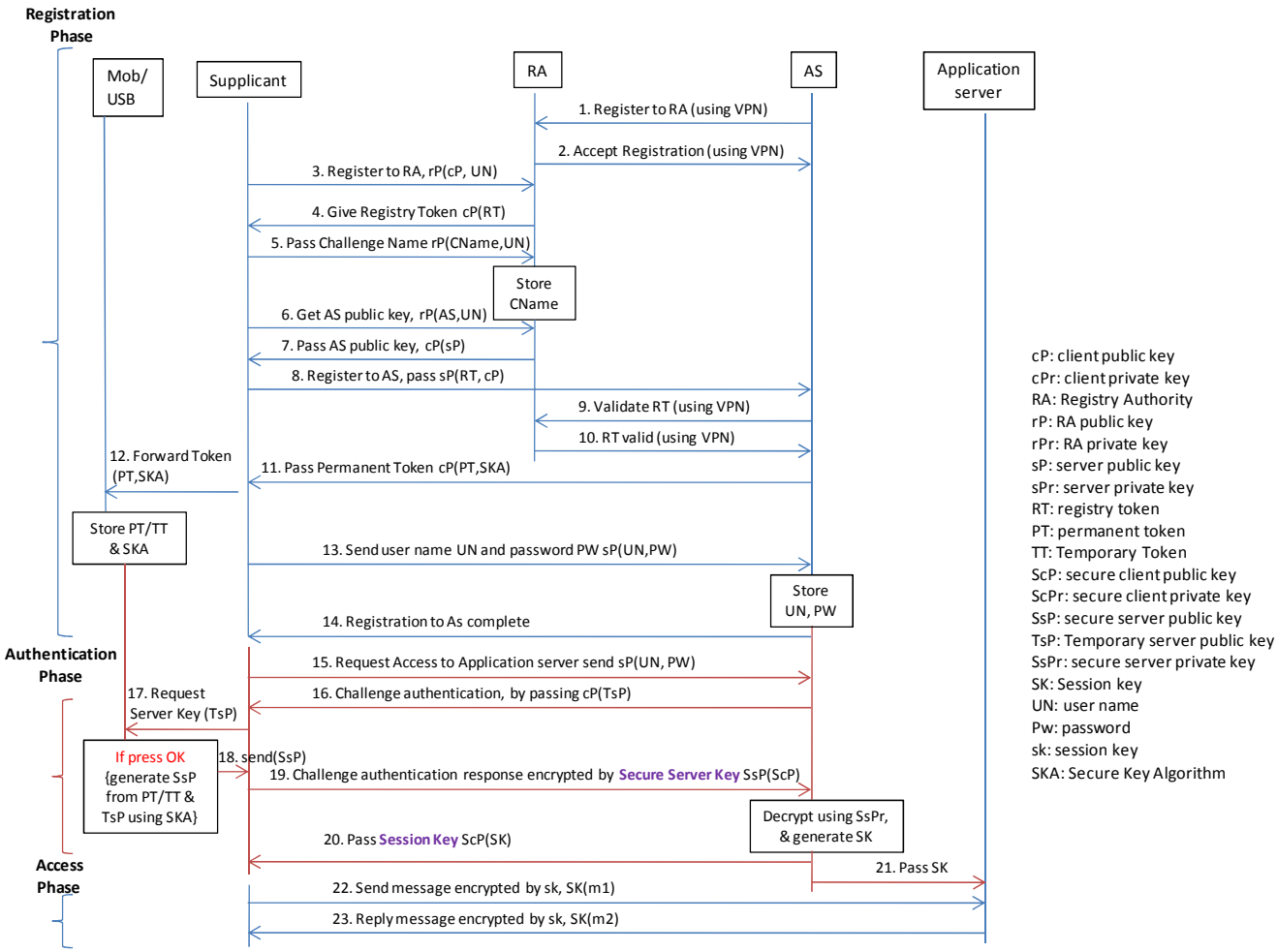

Figure 2. Challenge Token-Based Authentication process at different phases

III. SECURE KEY ALGORITHM TO LINK PUBLIC KEYS WITH PERMENANT STORED TOKEN

During supplicant's registration with the AS, the supplicant will receive from AS a Permanent token PT as well as a Secure Key Algorithm (SKA) which correlates this token to a secure AS server public key SsP. This will assist the supplicant to generate the SsP later in respond to a challenge request sent by the AS to validate the supplicant.

During authentication, if the registration token is valid, user can submit a user credential which will be saved on the AS, and then will obtain a Permanent Token (PT) from the AS. A generated SsP would be used instead of normal authentication public key that is known to everyone to exchange a session key between the supplicant and the application server.

The following algorithm describes the correlation between the SsP and PT:

X = PT {Permanent Token: stored in client side during registration}

Y = TsP {temporary server Public key: is a random challenge key sent to client to create a Secure server public key SsP}

Z = SsP {Secure server Public key: is a random server public key linked to TsP and PT through a KSA algorithm and never be transferred to client}

$Y = f(X, Z)$,

$Z = f(X, Y)$

As example, by using the algorithm: $X = 10*Y + 20*Z \rightarrow Y = (X - 20*Z)/10$, $Z = (X - 10*Y)/20$

For $Y = 200$, $Z = 500 \rightarrow X = 2000+10000 = 12000$

If AS sends $Y = 200$ to client, and client knows $X = 12000$, then client calculate $Z = (12000 - 2000)/20 = 10000/20 = 500$

If the AS server is able to decrypt a message that is received from the supplicant using its own private key SsPr, it will extract the user name and password. *A Successful*

decryption is an indication that client used the same algorithm (by using PT (X) and TsP (Y)) to generate the SsP (Z).

This secure server public key SsP would then be used to send the user credential over a secure session, and once the server verifies the user credential and approve it, it sends a session key to the suppliant and the application server APS to enable them exchanging messages using the same session key SK.

A program that is under development for the supplicant side is intended to generate the SsP in response to the challenge of the server by correlating the received TsP with the stored SKA. The SsP will be used by the supplicant to encrypt the response that will be sent to the server, while the server would be able to decrypt the response and authenticate the supplicant.

Using the secure server public key SsP instead of a well known server public key used in other techniques such as the one used in WAP, would avoid the DoS attack that will try to bring down the AS. The session key will be generated by the AS following a successful authentication of supplicant, and would then be transferred to supplicant using a secure client public key ScP.

IV. IMPLEMENTATION ISSUES AND EVALUATION

The two factor authentication would strengthen the security, where a hardware token or software token can be used for this purpose. Hardware Tokens are small devices that are carried by customers, where these tokens usually store cryptographic keys or biometric data. They are used mainly to display a Personal Identification Number (PIN) that changes with time, where customers/users use the PIN displayed on the token in addition to the normal account and password during the authentication. There are several commercial two factor authentication systems using the hardware token [15, 16]. Many banks offered the use of hardware tokens to their customers such as Bank of Queensland, the Commonwealth Bank of Australia and the Bank of Ireland [10]. The hardware token has its own limitation where customers that uses more than one two-factor authentication system requires carrying multiple tokens/cards which are likely to get lost or stolen. In addition, the hardware token solution is costly, where organizations such as banks with a million of customers, have to purchase, handle and maintain a million of tokens, as well as training their customers on how to use the token. Moreover, there is additional cost associated with stolen, lost and broken tokens.

Software tokens on the other hand overcome the limitations associated with the use of hardware tokens, since the device holding the tokens can store multiple tokens which can be used for different authentication applications, and the cost is minimized since the same device can handle multiple tokens. Software tokens are programs that run on computer/device, where they can provide a PIN that change on time such as the One Time Password (OTP). Mobile phone can be used for storing the soft token [14], where it uses a client program to generate OTP locally. However, if the mobile is out of sync with the server, it has to send a SMS message requesting the server to generate OTP and send it back to the mobile. This system might not be reliable if mobile network does not exist or it is out of coverage, and can be costly if it was out of sync where at least 2 SMS will be send to obtain the OTP.

In the proposed CTA method described in this paper, there is no need to send SMS messages in order to complete the authentication. In CTA, the supplicants respond to a challenge coming from the AS through the generation of a secure server public key (SsP) if the mobile phone can connect to the supplicants using WIFI or Bluetooth as shown in Figure 1, or through USB direct mobile connection in the absence of WIFI or Bluetooth connection. In both cases the stored secure key algorithm and to the stored software token are used to generate the SsP. A program that is under development for the supplicant side is intended to generate the SsP in response to the challenge of the server. This supplicant authentication (SA) program is a web interface, where if users want to login into a secure site using the CTA method, this program detects if WIFI or Bluetooth is enabled at the supplicant device so it passes the username and password entered by the users and waits for a challenge response from the server as shown in Figure 1. Once the challenge response is received from the AS server, it forwards it to the mobile phone, where the Mobile Authentication (MA) program will extract the TsP from the packet sent by the supplicant and generates the SsP using the stored permanent token and SKA, and then send it to the SA program. The SA program uses the SsP to encrypt and send the challenge response to the AS server. However, if the SA program did not detect any WIFI or Bluetooth on its device, users can connect the mobile phone to the device using USB connection and the MA program will interact with the SA program to complete the authentication using the same procedure described above.

V. OBTAING A LOST REGISTRY TOKEN THROUGH REGISTER AUTHORITY

Figure 3 presents the different scenarios for obtaining key from RA for the CTA as follows:

- It can register in a Register Authority RA every time you want to use a new machine so it can store the registry token obtained from RA on the new machine or a mobile phone or USB as shown in Figure 3.a.
- You can store the registry token at mobile phone or secure USB using finger print to avoid losing it and allow the authenticated user to access it from the USB as shown in Figure 3.b.
- Supplicant can complete the registration to AS by obtaining the AS public key through RA as shown in Figure 3.c
- If supplicant uses a public machine or it lost the RT, it can obtain the RT from the RA, and after completing the registration, the RT will be stored in mobile phone of USB, and if it uses a public terminal then it should delete all the sessions. The request can be done by submitting a challenge name CName to RA, as shown in Figure 3.d.

 ISBN: 978-1-61208-140-3

VI. CONCLUSION

In this paper a new technique "Challenge Token-based Authentication - CTA" has been presented, where, CTA uses 2 authentication factors, a normal user name and password authentication, and a challenge token based authentication. The first authentication used everywhere in current e-transaction applications, which is not secure enough since user name and password can be cracked especially in WLAN. The second authentication method CTA would make the authentication mechanism much stronger by using the soft token stored in a mobile phone or USB to generate a secure server public key SsP following the challenge request/response between supplicants and AS. Since the second authentication mechanism uses a token and a secure key algorithm that will never be transmitted during authentication, the derived SsP would be highly secure. The SsP will be used for exchanging the session key SK which is used for encrypting/decrypting the messages.

The use of SsP instead of known AS server public key will prevent any users from communicating with the AS unless they have a valid token and be able to generate the SsP. Consequently, hackers will not be able to use the cracked user name and password to access the AS, and they cannot launch a Denial of Service (DOS) attack on the AS since the SsP is only known to valid supplicants. The proposed CTA two factors authentication would eliminate the need for sending SMS messages that are used currently in the mobile second authentication techniques and therefore saving cost. Furthermore, the second authentication factor will eliminate hackers who know the User Name and password from accessing the application server since they cannot know the token and they would fail in generating the SsP. Implementing the mobile client program to generate the SsP using the secure key algorithm is in progress.

REFERENCES

[1] J. Geier, "802.1X Offers Authentication and Key Management" *http://www.wi-fiplanet.com/tutorials/article.php/1041171/8021X-Offers-Authentication-and-Key-Management.htm,* May 7 2002, accessed 17 April 2011.

[2] J. Geier, **"**Wi-Fi Protected Access (WPA)", *http://www.wi-fiplanet.com/tutorials/article.php/2148721* **,** March 20 2003**,** accessed 17 April 2011.

[3] *http://grouper.ieee.org/groups/802/11/*, July 18 2004

[4] J. Rodrigues, "SECURITY FOR THE WIRELESS NETWORK" WatchGuard Technologies, http://www.docstoc.com/docs/14980698/Wireless-Network-Security Inc, 2006-2007, accessed 17 April 2011.

[5] IEEE 802.11 "Wireless LAN Security with Microsoft Windows *Microsoft Corporation"* http://www.microsoft.com/downloads/en/details.aspx?FamilyID=67fdeb48-74ec-4ee8-a650-334bb8ec38a9&displaylang=en *Published: January 2008,* accessed 17 April 2011.

[6] G. Kbar, "Wireless Network Token-Based Fast Authentication", IEEE ICT2010 International Conference on Telecommunication, Doha Qatar 4-7 April 2010, pp. 309-315.

[7] B. Schneier, "Two-Factor Authentication: Too Little, Too Late," in Inside Risks 178, Communications of the ACM, 48(4), April 2005.

[8] V. Griffith and M. J. Messin: "Deriving mothers maiden names using public records". In J. Ioannidis, A. D. Keromytis, and M. Yung, editors, Applied Cryptography and Network Security (ACNS), pp. 91–103. Springer-Verlag, 2005. LNCS no. 3531.

[9] J. Brainard, A. Juels, and R. Rivest, "Fourth-Factor Authentication: Somebody You Know", CCS'06, , October 30–November 3, 2006, Alexandria, Virginia, USA. pp. 168-178, Copyright 2006 ACM1-59593-518-5/06/0010

[10] D. de Borde, "Two-Factor Authentication," Siemens Enterprise Communications UK- Security Solutions, 2008. Available at http://www.insight.co.uk/files/whitepapers/Two-factor%20authentication%20(White%20paper).pdf, accessed 17 April 2011.

[11] A. Kemshall and P. Underwood, "SecurEnvoy -white paper" July 2007, http://www.securenvoy.com/WhitePapers/white_paper_two_factor_authentication.pdf , accessed 17 April 2011.

[12] Zhangjiajie, "Mobile Authentication Scheme Using SMS", IITA International Conference on Services Science, Management and Engineering, July 11-12 2009, pp. 161-164, ISBN: 978-0-7695-3729-0

[13] M. Agoyi and D. Seral, "SMS SECURITY: AN ASYMMETRIC ENCRYPTION APPROACH", Sixth International Conference on Wireless and Mobile Communications ICWMC.2010.87, 20-25 September 2010, Valencia, Spain, pp. 448-452,

[14] F. Aloul, S. Zahidi, and W. El-Hajj, "Two Factor Authentication Using Mobile Phones", The 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-2009), Rabat, Morocco, May 2009, pp. 641-644.

[15] BestBuy's BesToken, http://products.esecurityplanet.com/security/identity/BestBuy-Deluxe-BESTOKEN.html , accessed 22 April 2011.

[16] RSA's SecurID, http://www.rsa.com/node.aspx?id=1156, accessed 22 April 2011.

 ISBN: 978-1-61208-140-3

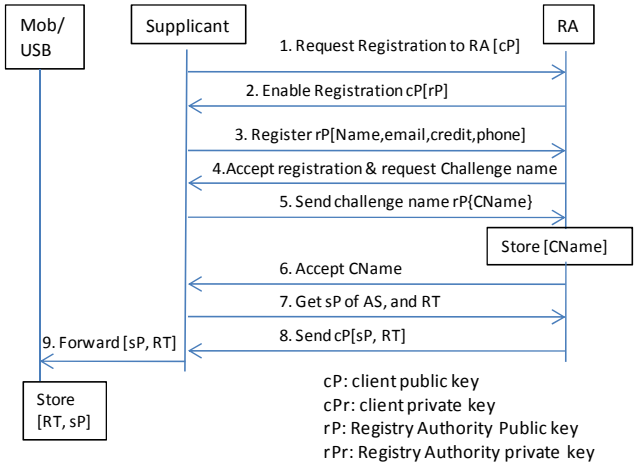

Figure3-a. Obtain Registry Token and AS server public key through Registry Authority through registration

Figure3-b. Obtain Registry Token and AS server public key through secure USB

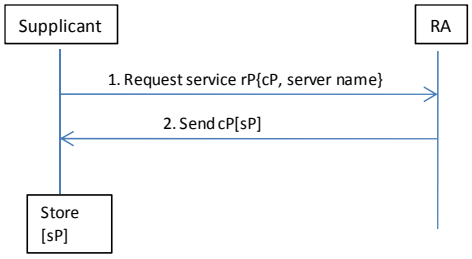

Figure3-c. Obtain AS server public key through Registry Authority after registration

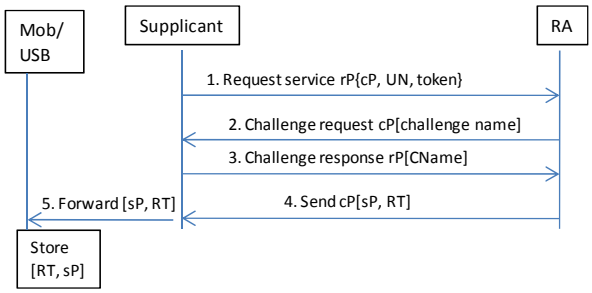

Figure3-d. Obtain lost/un-stored Registry Token and AS server public key through Root Authority after registration

Figure 3. Different scenarios for obtaining the token

 ISBN: 978-1-61208-140-3