Trusted Anonymous Authentication Scheme for Trusted Network Connection in Mobile Environment

Wu Jun-jun, Fang Ming-wei*, Zhang Xin-fang, Wang Tong-yang
Research Center of Software & Embedded System Engineering Technology, Huazhong Univ. of Sci. & Tech., Wuhan 430074, China
*Email: charlesfang@smail.hust.edu.cn

Abstract—Technologies make the mobile terminals such as smart phones, PDAs and handsets much more powerful to access mobile network in recent years. Especially with the widely use of mobile terminals, mobile network now becomes a primary tool for daily and business interactions. However, the proliferation of mobile terminals also draws mobile malware’s attention which will do damage to the mobile terminal and further affect the security of mobile network. But the traditional access control and authentication mechanism cannot resolve such security issues. On the basis of trusted computing technology, we proposed a mobile trusted network architecture by extending the trusted network connection in mobile environment. And an improvement EAP-EHash method is used in the proposed architecture to implement authentication. We defined two service scenarios in the authentication scheme, home network authentication and roaming network authentication. The process of each scenario is described in detail. By introducing the pseudonym mechanism, our scheme can protect user identity. And the connection status not only depends on the identification process, but also the trust status of the platform. The analysis shows that our scheme benefits the properties of user identity anonymity, mutual authentication, fake agent resistance, platform integrity verification, EAP and TNC Compatible. And the cipher suite negotiation makes our scheme more suitable for resource limited mobile terminals.

Index Terms—trusted computing; mobile network; trusted network connection; authentication

I. INTRODUCTION

With the rapid development of mobile communication technology and the popularity of smart mobile terminals, mobile network has become indispensable in people’s daily lives. The mobile network makes our lives colorful; however, it also leads us to new security challenges. As the wired network and internet become the main way of spreading and attacking of malicious code now, the development of mobile technology and service will also make the mobile network and terminals potential targets for malware and hackers in the similar way [1]. The spreading of malicious software in mobile terminal will become an increasingly serious problem, such attacks include: physical damage to terminal hardware, loss of information, unsolicited information and the denial-of-service against mobile networks. It can be expected that, while the malware propagates via the network connection of compromised terminals, it can widely spread to worldwide in a short period of time, which would cause more serious harm.

From the perspective of information security, security issues of internet are mainly originated from the terminal device which connected to the network. Therefore, in order to prevent the malicious attacks on the mobile network, the compromised device should be restricted from accessing to mobile network. When accessing the mobile network, the following security requirements are needed: confidentiality, device authentication, integrity, non-repudiation, availability, reliability and network control. As the mobile network security situation are increasingly serious, traditional network technologies, such as VPN, firewall, anti-virus software, instruction detection systems and NAC, are not able to meet these security requirements. The VPN mainly uses encryption to establish a temporary secure network connection via the unsecure public network. It can ensure user authentication and the confidentiality of the transferred data, but fails in determining the trust state of the connected device. In this situation, a compromised device may still attack the network which it connects to by using VPN. Network Access Control (NAC) [2] is a new security concept that was developed in recent years. It provides dynamic network access according to predefined security policy, and mitigates attacks within a network by managing the terminal’s connection to the network. The current generation of NAC mainly prevents unauthorized users from accessing the network by using the authentication mechanism to determine a user’s role. The same as VPN, NAC cannot analyze the trustworthiness status of the accessing terminal system either.

In order to resolve the security issues existed in network access, The Trusted Computing Group defined...
an open NAC standard called Trusted Network Connect [3] based on trusted computing. TNC achieves network control by collecting the trustworthiness of the terminal which depends on the overall state information of the terminal software and hardware configuration. When a device tries to access the network, first of all, user identification is accomplished, and next, the TNC components will check the status of the hardware and software configuration information. Then the trusted measure results will compare to the pre-defined security policy. If identical, the device will be allowed to access the network; otherwise the terminal will be rejected or isolated. Now many researchers pay their attention to using TNC technology to resolve the security issues in network. [4] proposed a TNC compatible network access control system, which enforces the connection state of the endpoint by determining the endpoint integrity and security status. The AES algorithm and the Diffie-Hellman secret key exchange protocol are used to implement platform authentication and secure communication. In [5], a trusted level division function is introduced to extend the trusted attribute to the trusted grade, in this way a hierarchical trusted network is constructed. And they have implemented hierarchical manageable access control and feasible remediation services in the framework. [6] pointed out that the use of NAC currently limited in local area network and VPN environment. By proposing a new protocol stack, authors extended the application of TNC to the web environment. By analyzing and comparing the existing network control technology, integrating MTM (Mobile Trusted Module) as root-of-trust on smart phone, [7] extended the TNC to the mobile platform in order to resolve the security issues in mobile network. The proposed VOGUE architecture connected mobile platforms and management server by using the VPN gateway, and it granted the access control rights to the terminal depends on the comparing results of the integrity measurement results in mobile platform. In the assessment process of TNC, the user may be reluctant to submit privacy information to the TNC server. Moreover, there exist threats of disclosure of standard IF-M data and arbitrary data in the TNC specification, which may expose user privacy. By adding a new policy manager to control the measured and verified integrity information in the client side, [8] proposed a privacy preserved trusted network connect scheme. For the trustworthy problem in TNC, [9] applied a Semi-Markov process based security quantifying method to analyze the TNC protocol in three aspects: authenticity, integrity and confidentiality. And they proposed several security enhancement mechanisms to address the threats and vulnerabilities during the integrity measurement and access authorization process in TNC architecture.

Although the TNC architecture is featured with security, authenticity, auditability, privacy, survivability and controllability [10], by analyzing the research status of TNC technology, [11] concludes that researchers should not only pay their attention to the core theory of trusted measurement and trusted report, but also constructing a theoretical model of TNC, along with the theory of data transferred mechanism and trusted resource sharing in trusted network. Meanwhile, there still exist several limitations in TNC, such as the one-way trusted authentication, message that transferred between components in the architecture lacks of the support of security protocols, and lacks of secure protection after accessing the network. For the mutual authentication issue in TNC, [12] uses EAP-TLS to implement mutual authentication between the access requestor and the access controller, and uses the trusted computing to implement trustworthy assessment of the platform. Such method is suitable for fixed networks, but may not extended to resource-constrained mobile environment. Due to the characteristic of mobile terminal, the authentication scheme for TNC should consume little computing resource in mobile platform, and roundtrips of the transferred message should be limited. More importantly, the method should be compatible to existing protocols.

II. TNC AND EAP METHODS

A. TNC Architecture

The Basic TNC architecture as depicted in figure 1, it consists of three entities, three layers and related protocol components. Compared to traditional network access control, two additional integrity measurement layer and integrity evaluation layer are added in the architecture to implement platform authentication and integrity verification when connecting to the network.

![Basic Architecture of TNC](Image)

Figure 1. Basic Architecture of TNC.

The three entities include access requestor (AR), policy enforcement point (PEP) and Policy Decision Point (PDP). The AR is in the client side, PDP can be treated as an AAA server. Both of them are generally composed of multiple software components, and each component is responsible for a specific function in the TNC architecture. When the network access requestor in AR sends a network connect request to the PDP via the IF-T interface, the integrity measurement collector will measure and collect the integrity measurement information of each component in the TNC client side, and then the TNC Client encapsulates that measured information and reports the platform integrity state to the TNC Server via the IF-TNCCS protocol. After receiving the request message from NAR, the Network Access Authority (NAA) in PDP will verify the identification of the platform. Next the integrity measurement verifiers
will verify the integrity status of the platform, and then it will determine the access status of AR, such as allowed, rejected and isolated, according to pre-defined security policy. In which, the IF-T protocol provides a secure authentication channel between NAA and NAR, and the data transfers between them are encryption protected by the session key. Finally, PDP sends the decision results to Policy Enforcement Point, which will enforce the access policies.

The three layers involve integrity measurement layer (IML), integrity evaluation layer (IEL) and network access layer (NAL). IML measures and collects integrity information of the platform, and then verifies the integrity state according to integrity report. IEL is responsible for the platform authentication and the assessment of integrity state of AR. NAL implements the secure connection of TNC Network by present network access technology, such as 802.1X, VPN and Point to Point.

B. EAP Methods

In the network access layer of TNC architecture, the authentication and data secure transmission between NAR and NAA are implemented via IF-T protocol. Existed Extensible Authentication Protocol (EAP) provides an authentication framework to support different EAP methods, which can not only be used for authentication, but also can transmit the integrity measurement information of terminal through the secure channel.

In order to reduce the computation resource consumption of terminals in the resource-constrained and vulnerable mobile network environment, the authentication procedure should satisfy following requirements: fast, lightweight and limited message transferred roundtrips. EAP-TLS and EAP-MD5 are the first two EAP methods that have been proposed. Recently, IETF promulgates six kinds of EAP methods (such as EAP-PAX, EAP-SAKE, EAP-PSK, EAP-POTP, EAP-FAST and EAP-generalized pre-shared key) and applies them to the mobile wireless network authentication. For the limitations of those method applied in mobile wireless network, [13] proposed an EAP-EHash mutual authentication method which can meet the requirements of the mobile wireless network. Such method is based on symmetric encryption algorithm, and can derivate robust session key. It has the properties as protection against Man-in-the-Middle attacks, and resistance to brute-force attacks and dictionary attacks. However, the EAP-EHash still has several limitations: it doesn’t provide identity protection in the authentication process, which will reveal the user privacy and lead tracking attacks.

III. SYSTEM OVERVIEW

Mobile network provides ubiquitous access services for users. The seamless roaming service which is not limited by the local home network is one of the most important services. That means, when the user is roaming, the terminal is still able to access the roaming network outside the local network, receive services and access resources.

The proposed trusted mobile network architecture is shown in figure 2. The main entities include mobile terminals, authenticator, issuer, home agent and roaming agent. A MTM is embedded into mobile terminal; it can provide basic cryptographic functionalities, such as random number generation, hashing, the security storage of sensitive data, asymmetric encryption and decryption, and signature generation, etc. The issuer is responsible for issuing DAA credential to the MTM. The wireless network access requestor in the mobile terminal sends the network access request, collects the integrity measurement value of MTM and then sends it to home agent by the authenticator. The authenticator, that is the access point, is the similar to the PEP in TNC architecture. The communication channel between mobile terminal and authenticator is wireless, and it is generally considered to be unsecure and vulnerable to eavesdropping, tampering, inserting and Man-in-the-Middle attack. The authenticator and the home agent are connected in the wired way, so the communication channel should be considered to be secure. The mobile terminal connects to the home agent via the authenticator. The home agent is the similar to the PDP in TNC, and it is mainly responsible for the registered user authentication and the integrity verification of the platform. After that, the home agent will provide service to the mobile terminal according to pre-defined policy. Due to the limited computation and power capacity of mobile terminal, an improvement EAP-EHash method, combined with trusted computing, is adopted to implement the authentication between the terminal and the home agent.

Figure 2. Architecture of Trusted Mobile Network.

There usually existed two service scenarios in the mobile network: the home network service and the roaming network service. In the home network scenario, user will provide trust proofs to the home agent at first, after successful verification, the user will register to the home agent, and then obtain the account and the shared key. After identification and platform verification implemented by the home agent, mobile terminal can connect to the home network directly. But in the roaming network scenario, there is no register information of the user in the roaming agent, so the mobile terminal cannot connect to the roaming agent directly. The roaming agent can only complete user identification with the help of home agent.
IV. AUTHENTICATION SCHEME

The proposed authentication scheme consists of the following processes:

A. Initialization

For one given security parameter 1, home agent (HA) chooses a multiplicative cyclic group G1 of large prime order p and an admissible bilinear map e: G1×G1 → G1. We let G1 ≃ G1. HA selects an integer x ∈ Zp as its private key skHA and computes its public key pkHA = x · g ∈ G1. Then HA selects three collision resistant hash functions, H1 : {0, 1}l → G1, H2 : {0, 1}l → Zp and H3 : {0, 1}l → {0, 1}l, where l is the length of the generated key. Meanwhile, HA selects a series of available ciphersuite CiphSu, and sets one symmetric encryption algorithm and one hash algorithm as the system default. Finally, the system public parameters are set to be < G1, G1, g, p, e, pkHA, H1, H2, H3, CiphSu >.

B. User Registration

Before registration, there exist two assumptions: first, the MTM generated the AIK (Attestation Identify Key) key pairs (skAIK, pkAIK) and obtains the AIK certificate CertAIK from the PCA. Second, the MTM has obtained the platform DAA credential cre ← (A, B, C) from a trusted issuer I. In the cre, A ← r · g, B ← y1 · A, C ← (x1 · A + rX1 · F), where r is a random integer selected by I, (x1, y1) is the private key of I, F ← f · g is a comment send to I, and f is the DAA secret of MTM. MTM computes D ← f · B, and securely stores cre and D in it.

When user want to register to Home Network, HA will make sure the binding relationship between the mobile terminal and the MTM at first. Mobile terminal selects two integers: h ∈ Zp and J ← G1, then computes A′ ← h · A, B′ ← h · B, C′ ← h · C, D′ ← h · D, and sets cre′ ← (A′, B′, C′). Next, mobile terminal computes c ← H1(cre′ || D′), then sends c and J to MTM. MTM first computes K ← f · J, then chooses one integer r ∈ Zp, and computes the flowing: R1 ← n · J, R2 ← n · B′, m ← H3(c || R1 || R2 || K || pkAIK), w ← n + m · f mod p. Finally, MTM sends (K, w, m) to the terminal. The mobile terminal sends (cre′, D′, K, J, w, m) to HA. When receiving the message, HA first checks whether the MTM is a rogue one by computing K = f · J, where f is a rogue DAA secret in rogue list. Then HA verifies whether e(A′, Y1) = e(B′, g) and e(A + D′, X1) = e(C′, g) equals, where X1 and Y1 are the public keys of I. If the above verification is successful, HA computes R1 ← n · J − m · K, R2 ← w · B′ − m · D′, c′ ← H3(cre′ || D′), then checks whether m′ ← H3(c′ || R1 || R2 || K || pkAIK) equals m. If identical, the binding relationship between mobile terminal and MTM can be guaranteed.

After the platform verification, HA generates an identifier IDu for the user, then generates the pre-shared key pskMTM ← x · QMTM for the mobile terminal, where QMTM ← H1(IDu || x), and constructs user personal account < Indu, Accu >, where Indu ← H2(e(QMTM, pkHA)) is the account index, Accu is the account detail information, which includes user identifier, consumer records, account balance and pre-shared key. Finally, HA sends pskMTM and the public parameters to the mobile terminal via secure channel. Mobile terminal securely stores the pskMTM in MTM.

C. Home Network Authentication

When the mobile terminal tries to connect to the home network, it should authenticate to the home agent, and negotiate for the session key. The home network authentication scheme consists of following steps:

Step 1: HA → MT: EAP Identity-Request

HA sends an EAP identity-Request to the mobile terminal, requests MT to send its device identity to HA.

Step 2: EAP Identity-Response

Mobile terminal selects t ← Zp, calculates R ← t · pkHA, and then transfers t and R to MTM. MTM generates a timestamp TMT, and computes the mobile terminal’s pseudonym PidMT ← (pskMTM · t−1)@ H1(TMT || R). According to the security requirements and the terminal configuration, user selects a proper ciphersuite CiphSu in the available ciphersuite. Otherwise the CiphSu is the default value. Then mobile terminal sends {R, PidMT, CiphSu} to HA as EAP Identity-Response.

Step 3: HA → MT: {IDuMT, RanduMT, EK, {MIC}, CiphSu}

HA will check whether the formula || TMT − TMT || latency is established or not at first, where the latency is the estimated network latency. Next, HA computes H1(TMT || R) using the received TMT and R, then retrieves (pskMTM · t−1) ← Pid @ H1(TMT || R) from the terminal’s pseudonym Pid, and the user account index can be computed by Indu = H2(e(pskMTM · t−1, x′ · R)). If this index exists, that means the user already registered. Otherwise the current user is unauthorized, the authentication protocol terminates. If registered, HA generates a random number RanduMT, and then derives the authentication key AK ← H1(pskMTM, R || RanduMT) and the encryption key EK ← H1(pskMTM, RanduMT || IDu || IDuMT) which is used to encrypt message integrity code (MIC) and hash values. Then, the MIC is computed to prove the integrity of the transferred message: MIC ← H1(AK, RanduMT || IDu || IDuMT || CiphSu). The MIC is encrypted using EK and the symmetric algorithm according to CiphSu in order to defend brute-force and dictionary attacks: EK{MIC}. Finally, HA sends {IDuMT, RanduMT, EK{MIC}, CiphSu} to the mobile terminal via authenticator.

Step 4: MT → HA: {IDuMT, RanduMT, Q, SIG, CiphSu}
While receiving the message from HA, mobile terminal will generate the authentication key AK and the encryption key EK, the same as HA, and checks whether the CipherSu is identical to the previous one. Then decrypts EK(MIC) using EK to retrieve MIC, and compares to the calculated MIC. If identical, the mobile terminal authenticates the HA successfully.

Next, MTM obtains the PCR (Platform Configuration Register) value of the current platform, as well as the corresponding storage measurement log (SML), and then generates a random number Randur. In response to the TPM_Certifykey command, MTM loads the AIK private key skAIK, and signs the platform configuration information with it: \( SIG \leftarrow Sig_{skAIK}(PCR, SML, Randur) \). Then the terminal generates the platform authentication message \( PAM \leftarrow H_1(\text{AK, SIG} \ || \ SML \ || \ Cert_{AIK}) \), and encrypts PAM with EK: \( Q \leftarrow E_K(PAM, SIG, SML, Cert_{AIK}, Randur) \). Finally, mobile terminal sends the authentication message to HA: \( \{ID_{MT}, Rand_{MT}, Q, SIG, CipherSu\} \).

**Step 5:** HA \( \rightarrow \) MT: EAP Success

HA \( \rightarrow \) Authenticator: \( K_i \)

After receiving the message, HA decrypts message \( Q \) by using EK, and retrieves platform integrity information \( SIG \), storage measurement log \( SML \) and AIK certificate \( Cert_{AIK} \). Then HA checks the integrity of \( PAM \) with authentication key AK. Next, HA will verify the integrity state of the mobile terminal platform with the help of the issuer. At first, HA validates the validity of the AIK certificate \( Cert_{AIK} \). If legitimate, HA retrieves the AIK public key \( pk_{AIK} \) from \( Cert_{AIK} \), and decrypts \( SIG \) to extract platform configure information PCR and Randur. Subsequently, HA evaluates whether the PCR value is compatible with the allowed security status. Depending on the evaluation result, HA determines the access status of the mobile terminal. If allowed and the first time access, HA generates a timestamp \( T_{HA} \), then sends the EAP Success message which contains \( T_{HA} \) to the terminal, and grants the mobile network access rights to it. Meanwhile, HA derives the master shared key \( MK \) with \( H_{1}(\text{psk}_{MTM},Rand_{MT} \ || \ Rand_{MT} \ || \ ID_{MT} \ || \ T_{HA}) \). In the legitimate time, \( MK \) only generates once. Subsequently, HA derives the session key \( K_i \) with \( H_{1}(MK, ID_{MT} \ || \ K_i \ || \ n) \), where \( K_i = ID_{MT} \), \( i = 1,2,\ldots,n \). Finally, the session key \( K_i \) which is used to encrypt transferred data is sent to the authenticator.

**Step 6:** While receiving the EAP Success message, mobile terminal will check whether \( MK \) is existed in the MTM or not. If not, the \( MK \) is derived the same as HA. And then the session key \( K_i \) is generated by using \( MK \).

**D. Roaming Network Authentication**

Due to the mobility of mobile terminal, users will roam to a foreign network at times. In this situation, the mobile terminal cannot connect to the roaming network directly due to lacking of register information of the user. In our roaming network authentication scheme, the roaming agent can authenticate the mobile platform and its platform integrity directly, and identifies user with the help of home agent where the user registered. Meanwhile, we assume the home network and the foreign network are in the same trust domain environment, the home agent and the foreign agent share the same issuer and possess each other’s certificate. The main idea of roaming network authentication can be concluded as following: the RA (Roaming Agent) sends an EAP identity-Request to the terminal. In response to the request, mobile terminal sends its pseudonym and platform integrity information to HA. RA verifies the platform credential and evaluates platform integrity state at first, then sends the signed trusted evaluation results and pseudonym to HA. HA authenticates the RA by using the RA’s certificate in order to defend counterfeit roaming agent attack, then identifies user by calculating the user account index according to pseudonym. The detailed authentication approaches are depicted in the following steps:

**Step 1:** RA \( \rightarrow \) MT: EAP Identity - Request

RA (Roaming Agent) sends an EAP Identity-Request to the mobile terminal, requests MT to send its platform integrity information and identity to RA.

**Step 2:** MT \( \rightarrow \) RA: EAP Identity - Response

While receiving the EAP Identity-Request from the Roaming Agent, mobile terminal retrieves the stored \( (cre', D', K, J, w, m) \) at first, then gets the platform PCR value and the corresponding SML value from MTM. Terminal triggers a TPM_Certifykey command to sign platform configure information with \( sk_{AIK} \): \( SIG \leftarrow Sig_{sk_{AIK}}(PCR, SML, Rand_{MT}) \). And the platform authentication message \( PAM \leftarrow H_1(SIG | SML | Cert_{AIK}) \) is generated according to the selected ciphersuite CipherSu, then mobile terminal encrypts \( PAM \) by using RA’s public key \( pk_{RA} \): \( Q \leftarrow E_{pk_{RA}}(PAM, SIG, SML, Cert_{AIK}, Rand_{MT}) \). Subsequently, mobile terminal calculates its pseudonym by using a timestamp \( T_{RA} \), a random integer \( t \) and \( R \): \( T_{RA} - t \cdot pk_{RA} = \text{PID}_{MT} \leftarrow E_{pk_{RA}}(\text{psk}_{MTM} \cdot t^{-1}) \cdot H_{1}(T_{RA} || R) \). Finally, mobile terminal sets \( \{ID_{RA}, Q, SIG, PID_{MT}, R, T_{RA}, CipherSu, (cre', D', K, J, w, m) \} \) as a roaming EAP Identity-Response message, and sends the EAP Identity-Response message to RA.

**Step 3:** RA \( \rightarrow \) HA:

\( \{ID_{MT}, ID_{MT}, SIG_{MT}, PID_{MT}, R, T_{MT}, Rand_{MT}, CipherSu\} \)

After receiving mobile terminal’s EAP Identity-Response message, RA must complete the platform authentication to mobile terminal and sends user identify response to HA. RA retrieves the timestamp \( T_{MT} \) from the receiving message, and then checks the freshness of \( T_{MT} \) to defend replay attack. Subsequently, RA checks whether the MTM in the rogue list and determines the binding relationship between mobile terminal and MTM by using message \( (cre', D', K, J, w, m) \). After completing platform authentication successfully, RA validates the
AIK certificate, and then retrieves the PCR and $Rand_{str}$ to evaluate the platform integrity state. If the integrity state compatible with the security requirements, RA continues the remaining operations. Otherwise, it terminates the communication to mobile terminal. RA generates a timestamp $T_{str}$ and selects a random integer $Rand_{str}$, signs the message by using its private key $sk_{RK}$ and calculates: $SIG_{str} ← \langle \text{Sig}_{sk_{RK}}(T_{str}, Rand_{str}, ID_{str}) \rangle$.

Then RA sends $\langle ID_{str}, ID_{iha}, SIG_{str}, Pid_{str}, R, T_{str}, Rand_{str}, CiphSu \rangle$ to mobile terminal’s home agent according to $ID_{iha}$.

**Step 4**: $HA → RA : \langle ID_{str}, ID_{iha}, T_{str}, SIG_{str}, CiphSu \rangle$

While receiving the message from RA, HA first retrieves RA’s certificate depending on the receiving $ID_{str}$, extracts RA’s public key $pk_{RK}$ from $Cert_{RK}$, and then validates the signature on $SIG_{str}$ by using $pk_{RK}$ to implement authentication to RA. If the authentication is successful, HA calculates $(psk_{str}, r \cdot t) \leftarrow \text{Pid} \oplus H_{3}(T_{str} \ | \ | R)$ and user account index $Ind_{i} = H_{1}(e(psK_{str}, r \cdot t, x^{1} \cdot R))$. If the output index is existed, HA authenticates the mobile terminal successfully. Subsequently, HA derives the roaming master key:

$RMK \leftarrow H_{2}(psk_{str}, Rand_{str} \ | \ Rand_{str} \ | \ ID_{str} \ | \ ID_{str} \ | \ T_{str})$.

By using its private key, HA signs the message $SIG_{str} ← \langle \text{Sig}_{sk_{RK}}(ID_{str}, ID_{iha}, RMK, T_{str}) \rangle$ and sends $\langle ID_{str}, ID_{iha}, T_{str}, SIG_{str}, CiphSu \rangle$ to RA.

**Step 5**: $RA → MT : \langle ID_{str}, Token, Rand_{str}, SIG_{str}, T_{str}, CiphSu \rangle$

After receiving the message, RA verifies the timestamp $T_{str}$ at first, and then validates the signature on $SIG_{str}$ by using public key of HA. If the verification successful, RA will make sure the mobile terminal is a legal registered one in HA. RA generates an access token $Token$ for the roaming mobile terminal, and then derives the roaming session key $sk_{R}$ by $H_{2}(RMK, ID_{str} \ | \ Token \ | \ \eta)$. At last, RA sends $\langle ID_{str}, Token, Rand_{str}, SIG_{str}, T_{str}, CiphSu \rangle$ to the mobile terminal.

**Step 6**: While receiving the response message from RA, the mobile terminal derives the roaming master key $RMK$ the same as HA. Then verifies the signature on $SIG_{str}$ to determine whether the roaming agent is a counterfeit one or not. If not, mobile terminal will generate the roaming session key which is used for encrypting transferred data while roaming.

**V. SCHEME ANALYSIS**

We will analyze the proposed scheme to evaluate whether our authentication scheme satisfies the following security requirements, such as scheme security, platform integrity, user identifier anonymity and the improvements of EAP-EHash method. At last, we compare the scheme to existed mobile network authentication scheme:

**A. Scheme Security**

In the registration phase, the decision of binding relationship between mobile terminal and MTM is based on the computational complexity of discrete logarithm and the zero-knowledge proof, the security of which can be guaranteed. The pre-shared key $psk_{str}$ and the public parameters can be sent to mobile terminal via present security technologies such as smart cards and OTA.

All the keys are calculated by using the cryptographic hash function $H_{i}$. This decreases the possibility of being intercepted while transferred via the unsecure wireless communication channel. Due to the collision-resistance of the one-way cryptographic hash function, it is impossible to derive the same key without knowing the pre-shared key. Because of the irreversibility of $H_{i}$, even the attacker obtains the master shared key or the roaming master shared key, it cannot deduce the pre-shared key $psk_{str}$, so the security of the derived keys depends on the security of $psk_{str}$, which is securely stored in MTM. And the MTM is engineered to be tamper resistant.

In the home network authentication phase, the authentication to mobile terminal relies on calculating the use account index $Ind_{i}$ from transferred pseudonym. And the $Ind_{i}$ is computed based on the hash function and the bilinear map function, this will make impossible for unauthorized user to access the network. The authentication to the home agent is based on the pre-shared key and the message integrity code which is encrypted by $EK$. Because of the symmetric encryption, the brute-force and dictionary attacks to MIC are more difficult. So it is impossible for the mobile terminal to connect to a fake home agent.

In the roaming network authentication phase, roaming agent and mobile terminal are authenticating to each other all with the help of home agent. And the authentication between home agent and roaming agent are based on the mature PKI technology, whose security can be guaranteed. When the authentication is successful, home agent informs roaming agent and mobile terminal by its signature. If the verification of the signature successful, mobile terminal will knows the roaming agent is not a counterfeit one, and the roaming agent will make sure the user is registered in the corresponding home agent. After authentication, the mobile terminal accesses the network by using the token provided by the roaming agent.

All the timestamps in the scheme are used for preventing replay attacks.

**B. Platform Integrity**

In the registration phase, mobile terminal uses the DAA credential to prove it possesses an illegal MTM module in it and the MTM belongs to an illegal issuer group. That means the mobile terminal has the capability to report its platform integrity information to a remote agent. And in the roaming network authentication phase, in order to isolate the compromised mobile terminal, roaming agent will verify the trust status of the platform at first. The mobile terminal uses the DAA signature

© 2012 ACADEMY PUBLISHER
scheme to achieve this. The MTM retrieves the PCR value and SML, signs them with the AIK private key to prove the trust state. On the basis of MTM, mobile terminal extends the trust chain to the mobile network. And in this situation, the roaming agent will evaluate the trust reports according to the pre-defined policy.

C. User identity Anonymity

In our scheme, the user identity is contained in the user personal account \(< \text{Ind}_i, \text{Acc}_i >\), which is secure stored in home agent. When authenticating, a pseudonym \(\text{Pid}\) is calculated to replace the real user identity to prevent user identity exposure. The one who only knows the pre-shared key \(\text{psk}_{\text{at}}\) can calculate the \(\text{Pid}\), and it is impossible to deduce user identity from \(\text{Pid}\). The only way to get user identity is to compute \(\text{Ind}_i = H_2(\text{e}(\text{psk}_{\text{at}} \cdot t^{-1} \cdot x^1 \cdot R))\), which needs to know the private key of home agent. If the adversary intercepts the \(\text{Pid}\), without knowing the private key of home agent and the user account information, it still can not acquire the user identity. That will guarantee the anonymity of user identity.

Moreover, different mobile terminal has different pseudonyms, which is calculated depends on the random number \(R\) and the timestamp \(T_{\text{at}}\). So one mobile terminal cannot compute other’s pseudonyms rely on its own pseudonym. Due to the randomness and unpredictability of random number, the \(\text{Pid}\) is different every time, so the attacker cannot launch tracking attack to the user by the \(\text{Pid}\).

D. Improvements of EAP-EHash Method

Comparing to the original EAP-EHash method, our scheme is featured with several improvements. First, we add the identity protection mechanism in our scheme, and the user identity is replaced by the pseudonym \(\text{Pid}\) while transferred in the communication channel. Second, we simplify the ciphersuite negotiation phase. All the useable ciphersuite are declared as public parameters in initialization phase. Every mobile terminal can choose the suitable functions in the ciphersuite fields as index values \(\text{CiphSu}\) and then sends them to the agent. Third, we extend the EAP-EHash method to trust network connection. The authentication not only depends on the pre-shared key, but also the trust state of the mobile terminal. Fourth, the EAP-EHash method is extended to the roaming service scenario.

E. Scheme Comparison

As shown in Table 1, comparing to the traditional authentication scheme for trusted network connection based on EAP methods [12], our scheme benefits the properties of user identity anonymity, roaming authentication, fake agent resistance and ciphersuite negotiation. That means our scheme is more suitable for the roaming mobile environment. And the adoption of symmetric cryptograph meets the resource limitation requirements of mobile terminals such as limited battery capacity, limited computation power and limited memory capacities. Comparing the method in [14], our scheme proposes the most used home network authentication, which is more useful in daily life. And more importantly, our scheme is compatible with the existing EAP method and TNC architecture, and there needs little modification when use our scheme directly.

VI. CONCLUSION

In order to resolve the security issues in mobile network, a trusted anonymous authentication scheme for mobile trusted network connection is proposed in this paper. By making up the shortage of EAP-EHash method and combing the TNC technology, our scheme implements the home network authentication and roaming network authentication. This scheme achieves not only user identification in the case of user anonymity, but also the platform integrity verification, which aims at rejecting the compromised mobile terminal out of the mobile network. And moreover, our scheme can be directly used in existing authentication architectures due to the compatibility of EAP method and TNC technology. And the next step, we will implement the prototype in the mobile phone to simulate the whole authentication phase.

ACKNOWLEDGMENT

Instructions and supervisions from Professor Zhang Xin-fang, construction suggestion from Liu Min, and collaborations of Wu Jun-jun and Wang Tong-yang are gratefully acknowledged.

REFERENCES


Wu Jun-jun was born in Wuhan Hubei, 1972. He received his BS, MS and Ph.D. degrees in mechanical engineering from Huazhong University of Science and Technology (HUST). Currently, he is an Associate Professor in School of Computer Science and Technology, HUST, and a member of China Computer Federation. His research interests lie in information security, trusted computing and embedded system.

Fang Ming-wei was born in Xiaochang Hubei, 1984. He received his BS degree in Industrial Engineering from Huazhong University of Science and Technology. Now, he is a Ph.D. candidate in HUST. His research interests lie in trusted computing, information security, and mobile network.

Zhang Xin-fang was born in Wuhua Guangdong, 1965. He received his BS, MS and Ph.D. degrees in mechanical engineering from HUST. Now, he is a Professor in School of Computer Science and Technology, HUST. His research interests lies in information security, cloud computing and embedded system and applications.