Formal Verification of the Coordinated Robust Authenticaion Protocol for Wireless Networks

Khosrow Ramezani, Elankayer Sithirasenan, Vallipuram Muthukkumarasamy
Griffith University, School of Information and Communication Technology
Gold Coast, Australia
{k.ramezani, e.sithirasenan, v.muthu}@griffith.edu.au

Abstract— **Today we are moving towards a Post-PC era where people prefer to use hand held devices such as tablets or smart phones to maintain their presence in the E-communication world. Most of this communication is possible because of technologies such as Wireless Local Area Network (WLAN), WiMAX or LTE etc. Although the wireless hand held devices are capable of effectively connecting with their native network they often fail to maintain robust connectivity in unfamiliar networks. Therefore the need for collaborative and complementary usage has to be explored. With the range of approaches that are used to validate the mobile devices in heterogeneous wireless networks, the need for a user friendly, flexible and reliable method to interconnect and utilize the different classes of wireless networks is becoming inevitable. Wireless network users prefer to access different types of networks either independently or cooperatively. In either case, adequate security provision is critical to the successful operation of the networks. Moreover, emerging technologies should provide seamless migration between the varied networks. Hence the ability to use a single but unique set of credentials to authenticate the wireless devices in heterogeneous wireless networks would make the post-PC era more effective. In this paper we present details of the formal analysis carried out on our proposed coordinated robust authentication (CRA) protocol.**

Keywords-component; Wireless Networks, Authentication, Access Control, Heterogeneous Networks, Behavior tree, Formal Verification, Symbolic Analysis Laboratory, Linear Temporal Logict

I. INTRODUCTION

Today we are moving into a “post-PC” world! Not many people sit in front of custom built PCs to do their businesses any more. Hand held devices such as iPod Touch, iPhone, Galaxy S3, iPad, Galaxy Tab, Airbook, Notepad etc. are bringing in a new paradigm as to how people use and communicate information. These devices can be thought as a theoretical “black-box”. It is for people who want to use it without wanting to know how it works. When a new application is installed, the user sees the icon and starts using it. All this is happening and adopted rapidly because people are able to do a number of things without being restricted to one place. They can download apps, watch movies, listen to news, browse the web etc. while on the move.

However, the mobility of these post-PC devices is restricted to some extent due to the limitations in wireless data connectivity. A wireless device at home should preferably get its data connectivity through the wireless router, while on the move from the 3G or 4G network and while at work from the office wireless network. Hand held device users may also want to connect to other wireless networks when there is no 3G or 4G connectivity. To facilitate seamless connectivity, wireless devices must first get securely associated with the different wireless networks. However, due to the inherent nature of the wireless communications, wireless networks encounter numerous security problems compared to its wired counterpart. The most significant of these is the first time association. Whether it is a WLAN [1], WiMAX [2][2] or a 4G LTE [3][3], all wireless networks will have this setback. The lack of physical connectivity (anchor-attachment) from the wireless device to the network makes the wireless network more vulnerable and hard to protect against authenticity, confidentiality, integrity and availability threats [4][5]. Hence, to overcome this first time association problem wireless devices adopt a range of different techniques.

The Robust Security Network Association (RSNA) proposed in IEEE 802.11i [6][6] has emerged as the most popular method to counter the first time association problem. The RSNA technique is widely used in both WLANs and WiMAX. Although IEEE 802.11i security architecture offers sufficient protection to the wireless environment, it is up to the implementer to guarantee that all issues are addressed and the appropriate security measures are implemented for secure operation. A single incorrectly configured station could lead the way for a cowardly attack and expose the entire organizational network [7][8].

Notwithstanding the configuration issues, RSNA is the most preferred first time association method for wireless networks. The use of IEEE 802.1x port based access control [9] makes it more flexible for mutual authentication and key distribution. However, RSNA does not support seamless connectivity in heterogeneous networks. This forces the wireless devices to disconnect from one network and connect to the other network using different credentials. Hence, a wireless device will have to repeatedly authenticate itself as it roams from one network to another operators’ network, be it the same type of network or different. Therefore, a Coordinated Robust Authentication (CRA) Mechanism with the ability to use a single set of credentials with any network, wireless or wired would be of immense significance to both network users and administrators.

In this paper we first examine the various authentication mechanisms proposed by other researchers for similar applications and then illustrate our proposed mechanism. We then present details of the formal analysis carried out on the proposed protocol.

II. SURVEY OF AUTHENTICATION PROTOCOLS

Iyer et al. [10][10] assert that WLAN and WiMAX are particularly interesting in their ability towards mobile data oriented networking. They confirm that a scheme enabling mobility across these two would provide several advantages to end-users, wireless operators as well as wireless internet service Providers (WISP). Further, they propose a technique with a common WLAN/WiMAX mobility service agent for use across WLAN and WiMAX access. By incorporating an acceptable mapping mechanism between WLAN and WiMAX, they interface a WLAN Access Point with the WiMAX Access Service Network (ASN) gateway. The mapping function inside WLAN access point maps all 802.11 events to the R6 events. For example the event association request will be mapped to WIMAX pre-attachment request.

In their architecture the problem of handling mobility across WLAN and WiMAX boils down to the problem of handling mobility across WiMAX base stations that already have concrete solutions. Also, the mapping function consumes 1.82 seconds for EAP-TLS authentication in comparison to few milliseconds as shown in section VI. Further, their proposed architecture enables the same IP address to be used across both the WLAN and the WiMAX network interfaces, and keeps it seamless from an application perspective.

The EAP-FAMOS authentication method developed by Almus et at. [11][11] use the Kerberos based authentication in the existing EAP framework. OSNP is another EAP method based on Kerberos proposed by Huang et al. [12][12]. The protocol provides intra-domain and inter-domain authentication to a peer that already has its security association with the home network. The authors have proposed a hierarchal design for KDC servers with the Root KDC responsible for providing directory service to other KDC servers. In case of a request to a particular network other than the peer's Home network, the authentication server in the new network will obtain the authenticity of the peer from the home KDC. Although the authors suggest a quick password based authentication and roaming mechanism, they fail to provide details of the hierarchical design of KDC servers and the agreement between them. Moreover, all servers share a group key and in case of a key compromise, access points can masquerade as legitimate authenticators.

Apart from the high administrative costs in Kerberos based methods; their solution is mainly targeted at specific wireless networks and authentication mechanisms. Wireless service providers use different authentication schemes on their diverse types of wireless networks. For example, a WiMAX service provider may use the EAP-TLS authentication scheme on their custom AAA servers, whereas corporate entities may want to use EAP-TTLS authentication mechanism facilitating the use of their existing authentication databases such as Active Directory, LDAP, and SQL. Hence, for convergence of wireless networks it is significant to develop an authentication mechanism that is versatile and simple so that it can be effectively used in any type of wireless network.

Narayanan et al. [13][13] propose ERP, an extension to the EAP framework and an EAP key hierarchy to support Re-authentication. As specified in RSNA, MSK is generated on successful completion of the authentication phase (phase 2 of RSNA). Subsequently MSK is passed to the authenticator to generate the TSK (phase 3 of RSNA). The TSK is then used for data encryption between the supplicant and the authenticator. However, the EAP framework proposed by Narayanan et al. suggests two additional keys to be derived by all EAP methods: the Master Session Key (MSK) and the Extended MSK (EMSK) which forms the EAP key hierarchy. They make use of the EMSK for re-authentication and successive key derivations.

ERP defines two new EAP messages EAP-Initiate and EAP-Finish to facilitate Re-authentication in two round trip messages. At the time of the initial EAP exchange, the peer and the server derive an EMSK along with the MSK. EMSK is used to derive a re-authentication Root Key (rRK). The rRK can also be derived from Domain-Specific Root Key (DSRK), which itself is derived from the EMSK. Further, a re-authentication Integrity Key (rIK) is derived from the rRK; the supplicant and the authentication server use the rIK to provide proof of possession while performing an ERP exchange. After verifying proof of possession and successful authentication, re-authentication MSK (rMSK) from the rRK is derived. rMSk is treated similar to MSK obtained during normal EAP authentication i.e. to generate TSK [14][14].

Apart from the few modifications to the EAP protocol due to the introduction of two new EAP codes, ERP integrates with the existing EAP framework very well. To demonstrate the possession, supplicant uses rIK to compute the integrity checksum over the EAP-Initiate message. The algorithm used to compute integrity checksum is selected by the peer and in case of server's policy does not allow the use of cipher suite selected by the peer; the server sends a list of acceptable cipher suites in the EAP-Finish / Re-auth message. In this case the peer has to re-start the ERP process by sending the EAP-Initiate message and the integrity checksum using the acceptable cipher suites. Furthermore ERP also recommends use of IPsec or TLS to protect the keying materials in transit. However, EAP-ERP requires a full EAP authentication at first when a user enters a foreign network. Further, if one supplicant for any reason has not been able to extract domain name of the foreign network then it should solicit it from its Home server, this can result in long authentication delays.

Although a number of solutions have been proposed for providing seamless connectivity for wireless devices, our proposed method leverages the advantages of both RSNA and Public Key Infrastructure. In the following section we describe our proposed method.

III. ENHANCED CRA PROTOCOL

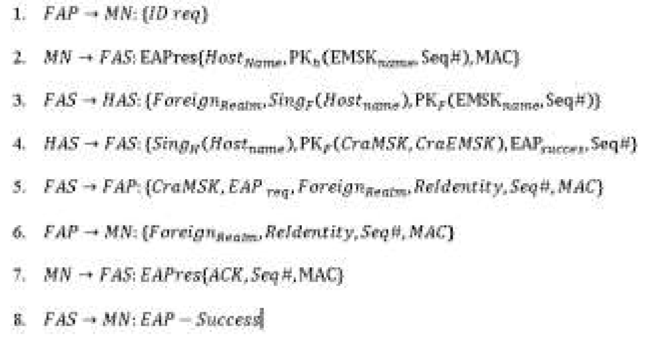
The principal notion behind the CRA mechanism is that every wireless device will primarily be associated with one wireless network, which can be referred as the HOME network. The credentials used by a wireless device to associate with its HOME network are assumed to be robust and specific to that network. Therefore, a wireless device must be able to use its authority in the HOME network to reliably associate with any other FOREIGN network. In this context, the AAA server that authorizes the wireless device in its home network is called as the HOME AAA Server (HAS) and the AAA server in a foreign network is called as FOREIGN AAA Server (FAS). Hence, our authentication mechanism will require only one set of credentials that it uses in the home network to access all foreign networks. We are considering both different types of networks and different authentication mechanisms that may be specific and effective to a particular type of network. Further details of our CRA protocol can be found in [15][15][16].

The Enhanced CRA protocol provides authentication in two modes; Full Authentication and Re-Authentication. With regard to mutual authentication CRA uses RADIUS servers as suggested in IEEE 802.1x [17]. RADIUS protocol exhibits better performance compared to other mutual authentication protocols [18][18]. CRA suggests direct communication between radius servers by pre-arranged agreement or the servers could find each other dynamically. In case the RADIUS servers do not have a pre-arranged agreement they can use their CA-signed PKI certificates to ascertain trust between servers.

All AAA servers that participate in the CRA must possess a CA-signed PKI certificate and be capable of obtaining the CA-signed PKI certificates of other participating AAA servers. Assuming that all AAA Servers that participate in the CRA are in possession of their CA-signed PKI certificates, the CRA protocol can communicate between the FOREIGN and the HOME AAA servers securely.

A.Full EAP-CRA authentication

Initial assumption of the CRA protocol is that each mobile device is primarily associated with a network, which in this context is referred to as the Home network. The security of the Home network and the authentication mechanism used must be robust. We anticipate that an EAP method such as EAP-TLS, EAP-PEAP or EAP-TTLS is used in the Home network. Therefore the values for MSKName, MSK, EMSK and the Time to Live (TTL) for these keys are available for the Peer. Since some of the EAP methods utilize CA-signed PKI certificates to authenticate and secure the communication, CRA extends it to add more flexibility to certificate based authentication. In this paper WLAN has been chosen as the medium to illustrate the components and messaging of EAP-CRA. Firstly, both the peer and the Foreign Access Point (FAP) discover their capabilities and decide on a suitable protocol to authenticate each other. If both parties are capable of EAP-CRA then the FAP will compose an EAP request message to solicit the identity of the peer. It should be mentioned that the key for hashing function is generated from the EMSK.



1. $FAP \rightarrow MN: \{ID\ req\}$
2. $MN \rightarrow FAS$: EAPres$\{Host_{Name}$, $\mathrm{PK}_h(\mathrm{EMSK}_{name}, \mathrm{Seq\#}), \mathrm{MAC}\}$
3. $FAS \rightarrow HAS: \{Foreign_{Realm}, Sing_F(Host_{name}), \mathrm{PK}_F(\mathrm{EMSK}_{name}, \mathrm{Seq\#})\}$
4. $HAS \rightarrow FAS: \{Sing_H(Host_{name}), \mathrm{PK}_F(CraMSK, CraEMSK), \mathrm{EAP}_{success}, \mathrm{Seq\#}\}$
5. $FAS \rightarrow FAP: \{CraMSK, EAP_{req}, Foreign_{Realm}, ReIdentity, Seq\#, MAC\}$
6. $FAP \rightarrow MN: \{Foreign_{Realm}, ReIdentity, Seq\#, MAC\}$
7. $MN \rightarrow FAS: EAPres\{ACK, Seq\#, \mathrm{MAC}\}$
8. $FAS \rightarrow MN: EAP - Success$

Fig. 1. Coordinated Robust Authentication (CRA) Full Authentication Protocol.

In an unknown network, the peer will first check if the TTL of MSK is still valid. Expired MSK will lead to a failed authentication and will prompt a full authentication. The peer will be responsible to do a full authentication with its Home network to obtain a fresh MSK. On the other hand, if the MSK is valid, the peer generates a random sequence number and encrypts the MSKname of home network and the sequence number by the public key of its HAS. The composed EAP-Response message will be sent to the FAP, which contains the encrypted message, Message Authentication Code, the realm of the home network and the random identity of the peer (message 2 in Figure 1).

FAP will encapsulate this EAP-Response message inside a RADIUS Packet and forward it to the foreign authentication server. The FAS will also utilize RADIUS for server-to-server communication. However before sending the received message, the FAP will add its domain name and encrypt the MSKname with its Private Key (message 3 in Figure 1). This enables the HAS to authenticate the FAS.

Upon receiving the message from a foreign network, HAS is able to check if the FAS is authorized based on the domain name of the FAS. The HAS can authenticate the FAS by verifying the contents of the signed message. Peer authentication will be managed by matching the MSKname with MSK, EMSK, Validation of key timer and the number of re-authentication of the peer. If the MSK is valid the HAS can combine the foreign domain name, sequence number and the previous EMSK to generate new CRA-MSK and CRA-EMSK.

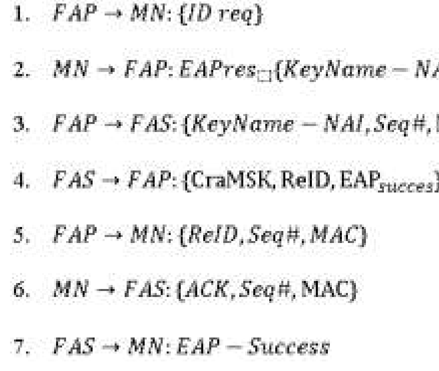
After updating the timer and counter values of the MSKname the HAS creates a RADIUS message which holds Access Accept, encrypted values of CRA-MSK and CRA-EMSK with FAS's Public Key, MAC and privately signed message of domain name – MSKname (message 4 in Figure 1).

FAS first checks the signed MSKname to validate the HAS, and then stores the MSKname and CRA keys. In addition to these it calculates a new timer, counter and random re-authentication ID for local re-authentication in case the peer stays for longer time in the foreign network. These values are CRA_timer, CRA_counter, and CRA_RND. The value of the CRA_timer must be less than the validity time of the initial MSK. Next, the FAS sends CRA_counter, re_id, EMSKname signed with HAS's private key, Foreign realm and CRA-MSK inside a RADIUS packet to FAP (message 5 in Figure 1). The CRA-MSK will be utilized for future communication to provide privacy. The rest of the message is sent to the peer (message 6 in Figure 1). The peer will be able to authenticate its home server by verifying the signature and generate CRA-MSK and CRA-EMSK. It then creates an EAP-Response as an acknowledgment with MSKname. The FAS can then compose an EAP-Success message and send it back to the peer.

On receiving the EAP success message, the peer generates rMSK independently leading to the key distribution phase. The key distribution phase will be similar to that of the RSNA where the supplicant and the authenticator will use the MSK to derive the Temporal Session Key (TSK). Once the TSKs are derived normal data communication can commence.

B. EAP-CRA Re-authentication

In the previous section we described a roaming-enabled authentication mechanism for users who wish to get connected to a new network, using the security credentials that they use in their home network. Although we anticipate relatively faster CRA authentication, in situations where the user continues to work on a foreign network the need for re-authentication is anticipated.



1. $FAP \rightarrow MN: \{ID\ req\}$
2. $MN \rightarrow FAP: EAPres_{\square}\{KeyName - NAI, Seq\#, \mathrm{MAC}\}$
3. $FAP \rightarrow FAS: \{KeyName - NAI, Seq\#, \mathrm{MAC}\}$
4. $FAS \rightarrow FAP: \{\mathrm{CraMSK}, \mathrm{ReID}, \mathrm{EAP}_{succes}\}$
5. $FAP \rightarrow MN: \{ReID, Seq\#, MAC\}$
6. $MN \rightarrow FAS: \{ACK, Seq\#, \mathrm{MAC}\}$
7. $FAS \rightarrow MN: EAP - Success$

Fig. 2. EAP-CRA Re-authentication

This section will explain the re-authentication process that can occur due to handover within the same network, i.e. when a user moves from one access point to another. The Enhanced CRA full authentication generates CRA-MSK and CRA-EMSK for a secure communication. Possession of these keys by the supplicant and the FAS can quicken the process of re-authentication. The FAS, after the successful authentication of a supplicant distributes the re-authentication identity and the CRA_Counter to the peer. The counter determines the number of re-authentications which can be acceptable.

The process of re-authentication will be initiated by the authenticator with EAP-Request for supplicant ID. In response the supplicant will check the time since last logon to verify the validity of CRA-MSK. In case the key is expired then a valid peer will fall back to request a full EAP-CRA authentication. On the other hand the supplicant sends its re-authentication ID and realm inside Kname-NAI, a random sequence number with a hashed value of the message. The key for the hash can be generated from the CRA-EMSK and sequence number. Here, the need for the sequence number arises to provide immunity against replay attacks. The authenticator will then forward the EAP-Response encapsulated as a RADIUS packet to the FAS (message 3 Figure 2).

Upon receiving the message the FAS checks the Kname-NAI with its stored authentication information. If there is a match, the server generates the hash value to verify the validity of the message and update the CRA_counter and CRA_timer values. The FAS will then send MSK, MAC, SEQ number to the authenticator. The authenticator retains the MSK and sends the rest to the peer. In the final step, the peer sends an EAP-Response as an acknowledgment. At this point the client is able to calculate the keying material, however to start secure communication the peer waits until it received the EAP-success from the authenticator.

Two sequence numbers, one with HAS and other with FAS are maintained for replay protection of EAP-CRA messages. The sequence number maintained by the supplicant and HAS is initialized to zero on generation of EMSK. The server sets the expected sequence number to the received sequence number plus one on every successful Re-authentication request, i.e. on generation of DSRK. Similarly, the supplicant and the FAS maintain a sequence number with the generation of rMSK while the supplicant is in the FAS's domain.

In the following section we present details of the formal verification carried out on our proposed protocol. We carried out the formal verification with the aid of Behavior Trees and the SAL environment. In the early stages of designing a protocol it is very difficult to verify its functionality, because on the one hand all aspect of the protocol are not determined and any implementation may be subject to significant changes. On the other hand simulations cannot be reliable unless the protocol is fully developed in the simulation environment. Therefore, we leveraged on the advantages of Behavior Trees (BT) that translates the requirements of a system to formal notations [19]. Formal verification of the proposed protocol is carried out using the SAL [20] environment with theorems derived using Linear Temporal Logic (LTL).

IV. FORMAL VERIFICAION

In this study we have used BTs to project the functionality of the proposed CRA protocol in a graphical form and to derive the SAL code for formal verification. We used Integrare [21] to draw the BTs and to generate the formal notation that is required for formal verification. First we translated requirements written in plain English in the form of BT representation with the aid of Integrare. Then all of the BTs were integrated to arrive at the integrated behavior tree. Formal representation of our proposed CRA protocol was then derived from the integrated behavior tree in the form of SAL code. Next, we developed a number of theorems in LTL and executed them on the formal representation (SAL codes) of our proposed CRA protocol. The theorems were tested against the SAL code using a Linux based program called SAL which can be accessed form http://sal.csl.sri.com [22].

A) Behavior Trees

Geoff Dromey [23] expresses behavior as a way that an entity functions. It can also be related to the way how two entities interact. This method provides traceable and clear association between requirements in ordinary language and their technical specifications. The behavior can entail specifications in the form of actions, result, events of even relations. For our purpose we just need three main states - a condition *?Condition?,* a static state *[state]* and data flow. To show data flow on the sender end the semantic is <*msg*> and on the receiving end is >*Msg*<.

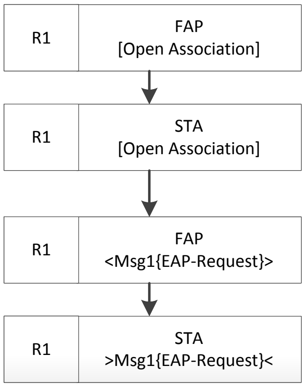


Figure 3. Graphical representation R1

The first step in designing behavior trees is to translate the requirements to functional requirements. The next step is to integrate the individual requirements to derive at the overall behavior tree. This step is the transition from requirements BTs to Design BT that will cover all behavior of the system. For example, consider requirement 1 (R1) that says "FAP sends a EAP-Request message to the supplicant". Requirement 2 (R2) says "the supplicant receives the message and verifies the capabilities of FAP for CRA authentication".

Figure 3 illustrates R1 – the FAP sends a "EAP-Request" to initiate the EAP authentication process. Figure 4 illustrates R2 – upon receiving Msg1 from the FAP, if the Supplicant is not capable of CRA it will disassociate otherwise it will continue to perform the authentication by validating the TTL value for the EMSK obtained from previous EAP authentication. If the TTL value is not expired the STA will compose a Message including its ID and domain name and send it to the FAP.

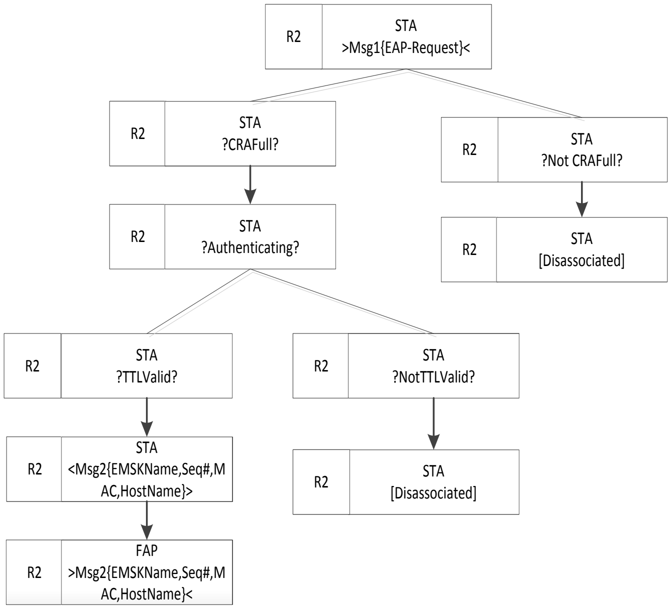


Figure 4. Graphical representation of R2

Next major step is the integration of requirements to generate the overall BT that incorporates all of the requirements. The integration between two requirements becomes possible if the post conditions of one requirement matches the pre conditions of another requirement. Considering R1 and R2, it is obvious that the post condition of R1 (STA receives the EAP-Request message) matches the pre condition of R2 (STA must receive the EAP-Request message). Figure 5 shows the integrated behavior of requirements R1 and R2. During the integration process the absence of an integration point reveals that there is either an ambiguity in the requirements or a requirement is missing altogether. Further, it may be that the requirement has not been translated correctly.

B) *SAL environment*

SAL stands for Symbolic Analysis Laboratory and is a static analysis tool for model checking and theorem proving. It uses a descriptive language to show the state transitions of entities. A detailed version of the translation of BT to SAL specification can be explored in [20].

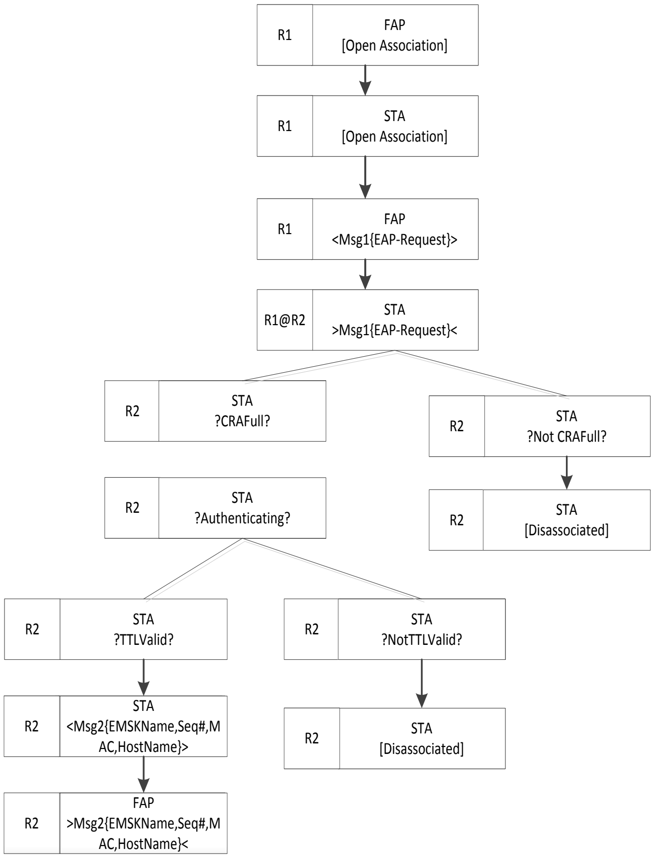


Figure 5. Integrated behavior of R1 and R2

SAL environment that was originally developed by David Dill of Stanford University and Thomas Henzinger of the University of California at Berkeley [20] has several tools such as *SALsim*, *SALenv* etc. It provides an environment to verify, simulate and check the functionality of a program. We used *sal-ssmc* to verify our temporal linear theorems against the SAL code generated from our BTs. The following LTL notations are used to compose our theorems:

- G(a): a is globally true, in other meaning it is always true.
- F(a) : states p will be eventually true.
- X(a): means next a happens

Partial listing of the SAL code for the integrated BT (R1 and R2) is given below:

```
BEGIN
LOCAL
fAP: FAP,
sTA: STA,
intInMsg_sTA_msg1:BOOLEAN,
intInMsg_STA_Msg1:BOOLEAN,
intInMsg_fAP_msg2:BOOLEAN,
intInMsg_FAP_Msg2:BOOLEAN,
pc1: [0..4],
pc2: [0..3],
pc3: [0..4],
pc4: [0..3],
pc5: [0..3]
INITIALIZATION
fAP=fAP_disassociated;
intInMsg_sTA_msg1=false;
intInMsg_STA_Msg1=false;
intInMsg_fAP_msg2=false;
intInMsg_FAP_Msg2=false;
pc1=1;
pc2=0;
pc3=0;
pc4=0;
pc5=0;
TRANSITION
[
A1:pc1=1-->
       sTA'=sTA_Disassociated;
       pc1'=2;
[]
A2:pc1=2-->
       intInMsg_sTA_msg1'=true;
       pc1'=3;
[]
A3:pc1=3 AND intInMsg_STA_Msg1=true-->
       intInMsg_STA_Msg1'=false;
       pc1'=4;
       pc2'=1;
       pc5'=1;
[]
............
A15:pc3=1 AND NOT (sTA=sTA_TTLValid)-->
       pc2'=0;
[]
A16:pc4=1 AND NOT (sTA=sTA_NotTTLValid)-->
       pc2'=0;
[]
A17: ELSE -->
]
END; % of MODULE
END
```

Once the system has been formally specified in SAL code, a number of analyses can be performed on the system specification. The sal-sim tool is a SAL simulator which is used to show different execution paths. The evaluation of different execution paths is usually a desirable first step to assess the correctness of the specification. The SAL specification can also be checked for deadlocks using deadlock-checker tool. The useful properties of the system can be specified using *linear temporal logic* (LTL) or *computation tree logic* (CTL). These properties can then be model checked using sal-smc and sal-bmc tools.

V. Analysis

In this section we discuss the results from the formal verification carried out on our proposed protocol. We analyze the important theorems that cover the main functionality of the protocol.

The first LTL formula has been defined to verify the normal behavior of the system. In other words a successful authentication must be the award for a legitimate user with valid credentials. As the theorem states if all conditions are true then finally the Supplicant will be authenticated.

```
th1: theorem behavior |-
       G((fAS=fAS_CRA)
       AND(fAS=fAS_CRAFull)
       AND(fAS=fAS_HASTrusted)
       AND(fAS=fAS_SignVerified)
       AND(fAS=fAS_MACValid)
       AND(hAS=hAS_FASTrusted)
       AND(hAS=hAS_MACValid)
       AND(hAS=hAS_NotKeyNameValid)
       AND(hAS=hAS_TTLVlaid)
       AND(hAS=hAS_SignValid)
       AND(sTA=sTA_CRAFull)
       AND(sTA=sTA_TTLValid)
       AND(sTA=sTA_MACValid))
       => F(sTA=sTA_Authenticated);
```

The above theorem was executed on our proposed protocol in the SAL environment. The theorem was proved without any counter example confirming that our protocol satisfies all of the conditions for normal operation. Once the model was verified for Normal behavior, we then verified the consistency of the security concerns. For this purpose we have considered the secure manner by which each component operates separately. To begin with we have three conditions which in all of them the system and the STA should act as expected. Theorem number two relates to the situation when the supplicant checks to see if the Key material from last EAP authentication is still valid or not. Logically if they are stale then the STA should disassociate.

```
th2: theorem behavior |-
       G(sTA = sTA_NotTTLValid)
       => F(sTA = sTA_Disassociated);
```

The next two LTL theorems are executed to confirm the behavior of the STA when the authentication protocol is not CRA while the initial request was for Full CRA and when the MAC value received in response to the first EAP-response to the authentication in not valid. In both scenarios the supplicant should move to the disassociated state.

```
th3: theorem behavior |-
       G(sTA = sTA_NotCRAFull)
       => F(sTA = sTA_Disassociated);

th4: theorem behavior |-
       G (sTA = sTA_NotMACValid)
       => F ((sTA =sTA_Discard) AND
              (sTA= sTA_Disassociated));
```

Theorem two, three and four were proved confirming that the STA will disassociate if the STA possesses expired key material, is not capable of EAP-CRA and has received a message with an invalid MAC. Next we derived at theorems 5 and 6 verifying the response of the system when either the HAS is not listed in the initial acceptable servers' list of the FAS or there is a false attempt by a counterfeit HAS. In both cases the FAS should Disregard the request and send failure message to the supplicant. In other words the Supplicant should be disconnected.

```
th5: theorem behavior |-
       G (fAS = fAS_HASTrusted)
       => X((sTA =sTA_Disassociated));

th6: theorem behavior |-
       G (fAS = fAS_NotSignVerified)
       => F ((fAS =fAS_Discard)
             AND(sTA = sTA_Disassociated));
```

Again theorems five and six were proved without any counter examples confirming that the Supplicant will be disconnected if the HAS is not trusted. Next, when the supplicant receives a second response from the FAP it means the STA has proved its identity and hence should be able to generate the key material. When the supplicant responds to the FAP, for security reasons it should be able to add a Message Authentication Code to the response message. Theorem 7 relates to this aspect - if the MAC is not valid then the STA will be sent a failure message and the will be disassociated.

```
th7: theorem behavior |-
      G (fAS = fAS_NotMACValid)
      => X((sTA=sTA_Disassociated));
```

The above seven theorems were model checked on the proposed protocol and proved to be true without counter examples. These are the minimum set of theorems that the protocol should satisfy for normal operation. As discussed above, in case there is a mismatch between the TTL timing values or the MSK name is not authentic in the home network or the MAC value is not valid at any point of time during the authentication process the system terminates the authentication request and forces the supplicant and the associated access point to a disassociated state.

The formal verification carried out on the proposed protocol in the SAL environment confirms the integrity, consistency and completeness of our proposal. In this process, we first validated all our assumptions using LTL theorems and thereafter derived theorems to prove the integrity of all state transitions. To further validate our proposed protocol we will be analyzing it against possible security threats in our future work.

VI. Conclusions

The need for a wireless hand held device to be constantly connected to the Internet is inevitable in this "post-PC" era. The hand held devices, as they roam from one wireless network to the other should be capable of both legitimately associating and uninterruptedly connecting with the new network. In this paper we have detailed an improved version of the CRA protocol that can facilitate the association of hand held devices in a foreign wireless network and have formally verified its overall functionality by model checking it in the SAL environment.

At its conceptual level the proposed protocol appears to be promising and rational. As the next step we intend to carry out a detailed security analysis of the protocol by identifying possible attack scenario and verifying the system response to such known security threats [24]. The success of the security analysis will guide us through to implementation.

References

[1] IEEE Std. 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[2] IEEE Std. 802.16-2004, IEEE Standard for Local and metropolitan area networks: Part 19: Air Interface for Fixed broadband wireless access systems.

[3] A. Ghosh, R. Ratasuk, B. Mondal, B. N. Mangalvedhe and N. T. Thomas "LTE-advanced: next-generation wireless broadband technology", in *IEEE Wireless Communications*, Vol. 17 , Issue 3, pp 10 - 12, Aug. 2010.

[4] C. He and J. C. Mitchell, "Security Analysis and Improvements for IEEE 802.11i", in *Proceedings of the 12th Annual Network and Distributed System Security Symposium*, NDSS 2005, pp. 90-110.

[5] A Perrig, J Stankovic, and D Wagner, "Security in wireless sensor networks", Wireless Personal Communications, vol 37, no 3-4, 2006.

[6] IEEE Standard 802.11i Part 11, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 6: Wireless Medium Access Control (MAC) Security Enhancements," July 2004.

[7] M. Lynn and R. Baird. Advanced 802.11 attack, Black Hat Briefings, July 2002.

[8] N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-Middle in tunneled authentication protocols. Technical Report 2002/163, IACR ePrint archive, United Kingdom, Cotober 2002.

[9] IEEE Std 802.1X-2001, "Local and Metropolitan Area Networks – Port-Based Network Access Control", June 2001.

[10] A.P. Iyer, and J. Iyer. "Handling mobility across WiFi and WiMAX", in *Proceedings of the 2009 international Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, IWCMC 2009, pp. 537-541.

[11] H. Almus, E. Brose and K. Rebensburg, "A Kerberos-based EAP method for re-authentication with integrated support for fast handover and IP mobility in wireless LANs", in *Proceedings of the 2nd international conference on communications and electronics,* ICCE 2008, pp 61–66.

[12] Y.L. Huang, P.H. Lu, J.D. Tygar and A.D. Joseph, "OSNP: Secure Wireless Authentication Protocol using one-time key", in *Proceedings of Computer and Security* 2009, pp. 803-815.

[13] V. Narayanan and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)," RFC 5296, Internet Eng. Task Force, 2008.

[14] J. Salowey, L. Dondeti, V. Narayanan and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)," RFC 5295, Internet Eng. Task Force, 2008.

[15] E. Sithirasenan, S. Kumar, K. Ramezani, and V. Muthukkumarasamy. "An EAP Framework For Unified Authentication in Wireless Networks". In TrustCom'11: *Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 92-99, Nov. 2011.

[16] E. Sithirasenan, K. Ramezani, and V. Muthukkumarasamy. "Enhanced CRA Protocol For Seamless Connectivity In Wireless Networks". In ISCIT'12: *Proceedings of the 12th International Symposium in Communication and In*formation Technology, pages 72-76, Oct. 2012

[17] L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," *RFC 3748, Internet Eng. Task Force,* 2004.

[18] Stanke, M., & Sikic, M. (2008). Comparison of the RADIUS and Diameter protocols. *Paper presented at the Information Technology Interfaces,* 2008. ITI 2008. 30th International Conference.

[19] E. Sithirasenan, S. Zafar and V. Muthukkumarasamy. "Analysis of IEEE 802.11i WLAN Security Protocol". *ISAST Transactions on Computers and Software Engineering*, vol. 2, no. 1, pp. 13 – 25, 2008.

[20] Moura, L., S. Owre, and N. Shankar. "The Sal Language Manual." SRI International Rev. 2, (2003).

[21] Wen, L., et al., "Integrare", a Collaborative Environment for Behavior-Oriented Design. Cooperative Design, Visualization, and Engineering, 2007: p. 122-131.

[22] Dutertre ,B .SAL 3.2 Binaries without Yices .Retrieved from http://sal.csl.sri.com/download.shtml , 2012.

[23] R.G. Dromey, "From Requirements to Design: formalizing the key steps", *Proc. 1st International Conference on Software Engineering and formal methods,* September 2003, pp. 2-11.

[24] E. Sithirasenan and V. Muthukkumarasamy. Detecting Security Threats in Wireless LANs Using Timing and Behavioural Anomalies. In *ICON '07: Proceedings of the 15th IEEE International Conference on Networks,* pages 66-71, November 2007.