Enhancing UMTS Authentication and Key Agreement with Vector Combination

Yaohui Lei, Samuel Pierre and Alejandro Quintero

Abstract

The Universal Mobile Telecommunications System (UMTS) represents an evolution in terms of capacity, data speeds and new service capabilities from second generation mobile networks. It also provides more secure wireless access security mechanisms. One of these mechanisms, the authentication and key agreement (AKA) protocol, is designed to help a foreign network to authenticate a roaming mobile user through authentication vectors generated by the authentication center (AuC) in the user's home network. However, AKA has been criticized due to its introduction of sequence numbers and its vulnerabilities of redirection attacks and active attacks in corrupted networks. Moreover, since only the home network can generate authentication vectors to its subscribers, the AuC actually becomes the traffic bottleneck. This paper proposes an enhanced AKA based on vector combination (VC-AKA) to eliminate the above drawbacks. Through vector combination, a size n array of authentication vectors can realize up to 2^{n-1} times mutual authentication instead of only n times in UMTS AKA. Hence, the traffic for the home network to generate authentication vectors is exponentially decreased. Also, VC-AKA abandons the employment of sequence numbers and no more re-synchronization procedure needed as in UMTS AKA. Through security analysis and comparison with related work, we show that VC-AKA is more efficient and secure.

Index Terms

authentication and key agreement, mobile communications, security, UMTS, vector combination

I. Introduction

THE unprecedented growth of world-wide mobile wireless markets, coupled with advances in communication technology and the accelerated development of services taking place in fixed

Y. Lei is with the Department of Computer Engineering, Mobile Computing and Networking Research Laboratory, École Polytechnique de Montréal, Montréal, Canada H3T 1J4 (e-mail: yaohui.lei@polymtl.ca).

S. Pierre is with the Department of Computer Engineering, Mobile Computing and Networking Research Laboratory, École Polytechnique de Montréal, Montréal, Canada H3T 1J4 (e-mail: samuel.pierre@polymtl.ca).

A. Quintero is with the Department of Computer Engineering, Mobile Computing and Networking Research Laboratory, École Polytechnique de Montréal, Montréal, Canada H3T 1J4 (e-mail: alejandro.quintero@polymtl.ca).

networks, instigates the emergence of third Generation Mobile Communication System (3G). The Universal Mobile Telecommunications System (UMTS) represents an evolution in terms of capacity, data speeds and new service capabilities from second generation mobile networks. It also provides more secure wireless access security mechanisms compared to the Global System for Mobile (GSM) Communications [1].

UMTS wireless security mechanisms has been widely applied while integrating UMTS with other networks [2]–[9]. Hence, providing secure wireless access mechanisms is critical to wireless network integration. One important mechanism in UMTS is the authentication and key agreement (AKA) [10]–[13]. UMTS AKA achieves mutual authentication by the user and the network through a long term shared secret key between the user's Universal Subscriber Identity Module (USIM) and the authentication center (AuC) in the user's home environment (HE). The design of UMTS AKA is closely based on the GSM challenge-response scheme. The mobile user's USIM maintains a sequence number with the user's home network. When the mobile user roams to a visited network, the home network sends a set of authentication vectors to the visited network. To authenticate the mobile station, the visited network picks up an unused vector and sends the challenge part to the mobile station. The mobile station checks the freshness of the sequence number and computes a response. Also, the mobile station computes the cipher key and the integrity key for further communication encryption. The visited network compares the response with the expected response in the authentication vector. If they are same, the mobile station is authenticated. Hence, mutual authentication is realized in UMTS. In GSM, only mobile station is authenticated.

However, UMTS AKA has been criticized due to its introduction of sequence numbers [14]–[16] and its vulnerabilities of redirection attacks and active attacks in corrupted networks [14]. A failure of sequence number synchronization could result in complicated protocol execution especially when the user roams to a foreign network [10], [15]. The redirection attack can redirect a legitimate user's traffic to an unintended network and also cause billing problem [14]. The active attack in corrupted networks may jeopardize the entire networks because an adversary can use the authentication vectors corrupted from one network to impersonate another network [14].

Another drawback of UMTS AKA exists in the distribution of authentication vectors. A size n array of authentication vectors can be used for n times authentication. Since only the HE is responsible for generating and sending authentication vectors for all subscribers, it actually becomes the traffic bottleneck.

In this paper, we abandon the adoption of sequence numbers and propose an enhanced AKA to eliminate the above drawbacks. The proposed AKA is based on vector combination: different vectors combining

together can also be used for authentication. Through vector combination, a size n array of authentication vectors can realize up to 2^{n-1} times mutual authentication instead of only n times in UMTS AKA. Hence, the traffic for the HE to generate authentication vectors is exponentially decreased.

This paper is organized as follows. Section II gives an overview of UMTS AKA. Section III analyzes briefly its weaknesses. Section IV presents the improved AKA based on vector combination. Then, security analysis is given in Section IV. A comparison with related work is shown in Section VI. Finally, we conclude this paper.

II. Overview of UMTS AKA

Fig.1 gives an overview of UMTS AKA mechanism. Detailed description can be found in [10]–[13]. The mobile station MS and its HE/HLR share a long term secret key K. This protocol describes the scenario that the MS roams to a visited location register (VLR) or a serving GPRS support node (SGSN). The communication channel between the VLR/SGSN and the HE/HLR is assumed secure [10].

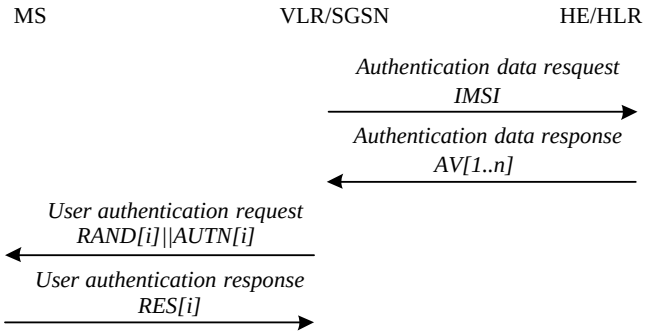

Fig. 1. UMTS AKA

Upon receipt of a request from the VLR/SGSN, the HE/HLR sends an array of ordered authentication vectors $AV[1..n]$ to the VLR/SGSN. The authentication vectors are ordered based on a sequence number. Each authentication vector consists of the following components: a random number $RAND$, an expected response $XRES = f2_K(RAND)$, a cipher key $CK = f3_K(RAND)$, an integrity key $IK = f4_K(RAND)$, an anonymity key $AK = f5_K(RAND)$, a message authentication code $MAC = f1_K(SQN \ RAND \ AMF)$ and an authentication token $AUTN = SQN \oplus AK \ AMF \ MAC$, where $f2_K$, $f3_K$, $f4_K$ are message authentication functions with the secret key K, AMF is the authentication management field. The authentication token $AUTN$ includes a sequence number SQN maintained between each pair of HE/HLR and the MS. Each authentication vector can be used for only one authentication and key agreement between the VLR/SGSN and the MS.

Then, the VLR/SGSN selects the next available authentication vector and sends the parameters $RAND$ and $AUTN$ to the user. The MS first checks the correctness of MAC in $AUTN$. Then, it computes the AK and retrieves the $SQN = SQN \oplus AK \oplus AK$. It verifies if SQN is in an acceptable range compared to the value he maintains. If it is not, he rejects the authentication and a sequence number resynchronization procedure may need to be executed by the HE/HLR. Otherwise, after successful verification of $AUTN$, the MS computes a response RES and sends back to the VLR/SGSN. The MS also computes CK and IK as the HE/HLR does. While receiving the response, the VLR/SGSN compares it with $XRES$ stored in the authentication vector. If they match, the VLR/SGSN considers the authentication and key agreement exchange completed successfully.

III. UMTS AKA Weaknesses

Although UMTS provides more secure AKA than GSM as claimed by 3GPP [10], there still exists several weaknesses. These weaknesses has been widely discussed in literature. We first give an overview of these weaknesses here and propose an enhanced AKA in the next section.

A. Sequence number: a bad choice

In UMTS AKA, one of the main drawbacks is the adoption of sequence numbers. The HE/HLR should keep and update a sequence number for every mobile user. The synchronization and re-synchronization involves complicated work (generation, allocation, verification, and management), especially with regard to the protection against an attack to force the sequence number wrap around and the compromise of user identity confidentiality [15]. A thorough analysis of operational difficulty with sequence numbers is given in [14]. Several scenarios which could result in synchronization failures are discussed in [14], [15]. Once a synchronization fails, the re-synchronization procedure involves quite a few entities and network domains [10]. Moreover, a possible crash in the database which stores the sequence numbers will cost a lot to re-establish the synchronization [14].

B. HE/HLR: a bottleneck

In UMTS AKA, the HE/HLR is responsible for generating authentication vectors upon receipt of requests from all VLRs/SGSNs. While the number of subscribers is usually large, the HE/HLR experiences heavy authentication traffic and actually becomes the bottleneck of the entire AKA scheme. This is even

worse if the mobile user roams to a far away foreign network as discussed in [17], [18]. Each work proposed an algorithm to investigate the traffic and determine the optimal number of authentication vectors.

C. Attacks in UMTS AKA

UMTA AKA is proved by BAN logic [19] in [10]. However, it has been suggested that the BAN logic is unable to deal with security flaws resulting from interleaving of protocol steps [20]. Recently, there are two attacks found in [14]: *redirection attacks*, *active attacks in corrupted networks*. The redirection attack occurs when an adversary entices a legitimate mobile station to camp on the radio channels of the false base station. Since the authentication vectors can be used for any serving network, an adversary can intercept the vector and impersonate both the mobile user and the serving network. Consequently, the adversary can redirect user traffic to an unintended network.

The active attack occurs while a network is corrupted and an adversary could forge an authentication data request from the corrupted network to obtain authentication vectors. In addition, the adversary could force the sequence number to be set in a very high value by flooding the authentication data to the home network. As a result, the adversary can start an active attack against all the legitimate users.

IV. VECTOR COMBINATION BASED AKA

In this paper, we propose an enhanced AKA based on vector combination (VC-AKA). In UMTS AKA, each authentication vector is used only once. After that, it should be abandoned in order to defend replay attacks. In fact, we observe that a combination of two vectors can also be used for authentication in condition that this combination should also be used only once. For example, suppose ($RAN\ D_1$, $X\ RES_1$) and ($RAN\ D_2$, $X\ RES_2$) are two challenge-response pairs of two vectors in UMTS AKA, then ($RAN\ D_1 \oplus RAN\ D_2$, $X\ RES_1 \oplus X\ RES_2$) can also be used for authentication since only the user who passes the single vector authentication can compute the correct response. Of course, extra works need to be done in order to defend replay and impersonation attacks.

In this way, a size n authentication vectors can have up to $(2^n - 1)$ combinations (including combinations with only one vector as in UMTS AKA). The HE/HLR can control the combinations allowed to be used or simply allow both the VLR/SGSN and the MS to use all combinations.

Based on this idea, we propose the protocol two procedures involved as follows. The first procedure is responsible for the distribution and establishment of authentication vectors; the second procedure realizes vector combination based AKA between the mobile station and the VLR/SGSN (see Fig. 2).

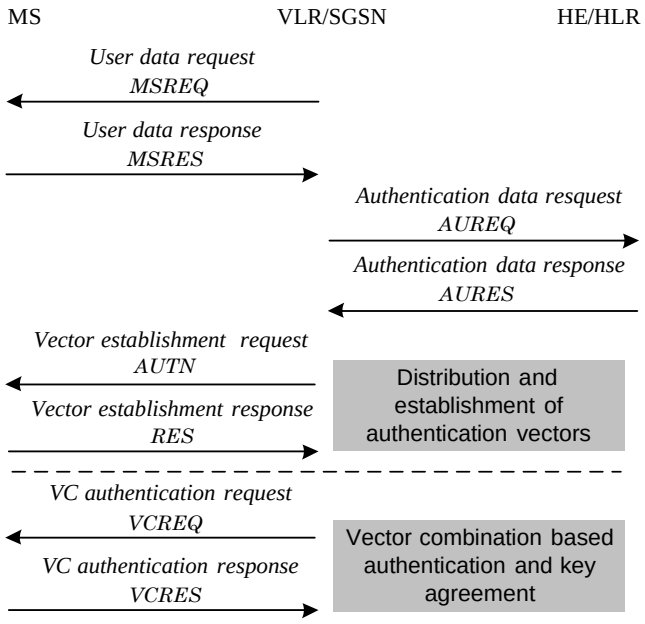

Fig. 2. Vector combination based AKA

When a mobile user roams to a visited network for the first time, the VLR/SGSN invokes the first procedure of this protocol by sending $MSREQ$ as *user data request* to the MS:

$$MSREQ = N_V$$

where N_V is a newly generated nonce. Upon receipt of $MSREQ$, the mobile station MS generates $MSRES$ as *user data response* and sends it back to the VLR/SGSN:

$$MSRES = V, H, N_M, MAC_M$$

where N_M is a nonce generated by the mobile station, and $MAC_M = f1_K(N_M \ N_V \ V \ H)$ is message authentication code generated by the function $f1$ with the secret key K. The MS indicates that V is the visited network and H is its home network for authentication.

Subsequently, the VLR/SGSN sends $AUREQ$ as *authentication data request* to the HE/HLR:

$$AUREQ = MSRES, IMSI, N_V$$

where $IMSI$ is the MS's International Mobile Subscriber Identity. Upon receipt of $AUREQ$, the HE/HLR verifies if the mobile user is a legitimate user and checks the correctness of the authentication message code MAC_M through the shared secret key K. If it fails, the HE/HLR rejects the authentication re-

quest. Otherwise, the HE/HLR succeeds the mobile user authentication. Then, it generates authentication data as follows. First, it generates random numbers R_x, R_y and $RAND$. It computes a response $XRES = f2_K(N_M \ N_V \ RAND)$, a session key $SK = f5_K(N_M \ N_V \ RAND)$ and an authentication token $AUTN = RAND \ N_M \ N_V \ \{R_y\}_K \ MAC$, where $\{R_y\}_K$ means encrypted R_y with the secret key K and $MAC = f1_K(RAND \ N_M \ N_V \ \{R_y\}_K)$. Then, it generates an array of n authentication vectors denoted by $AV[1..n]$. Each vector consists of a challenge-response pair $(RN_i, XRES_i)$, where $RN_i = f5_K(N_M \ N_V \ RAND \ i)$ is the challenge and $XRES_i = f2_K(N_M \ N_V \ RAND \ i) \oplus R_x$ is the response. Both of them rely strongly on the index i. The HE/HLR computes $R = f2_{SK}(R_x \oplus R_y)$ and constructs $AURES$ message as *authentication data response* shown in Fig. 2:

$$AURES = AV[1..n], R, XRES, SK, AUTN.$$

Upon receipt of $AURES$, the VLR/SGSN sends $AUTN$ as *vector establishment request* to the MS and stores the rest locally. The MS isolates the $RAND$, $\{R_y\}_K$ and checks the validity of the $AUTN$ by computing the MAC. If it is not valid, the MS rejects. Otherwise, the MS gets the R_y by decrypting $\{R_y\}_K$ with its secret key K. It computes the session key $SK = f5_K(N_M \ N_V \ RAND)$ and the response $RES = f2_K(N_M \ N_V \ RAND)$. Then, the MS generates an array of vectors $MAV[1..n]$. Each vector $MAV[i]$ consists of a challenge-response pair (XRN_i, RES_i), where $XRN_i = f5_K(N_M \ N_V \ RAND \ i)$ is the challenge and $RES_i = f2_K(N_M \ N_V \ RAND \ i) \oplus R_y$ is the response. It keeps SK and sends RES as *vector establishment response* to the VLR/SGSN.

Upon receipt of the RES, the VLR/SGSN compares if it is the same as $XRES$. If it is not, the VLR/SGSN rejects it and the protocol aborts. Otherwise, the first procedure finishes and authentication vectors are established in both the MS and the VLR/SGSN.

The second procedure is invoked when the VLR/SGSN needs to authenticate the MS. It sends *VC authentication request* to the MS. This request message is based on vector combination. The number of combinations for n vectors is $(2^n - 1)$ (at least one vector in each combination). Each combination, denoted by c, is a 0 and 1 n-bit array and can be used only once. The combination's decimal value is between 1 and $(2^n - 1)$. For example, the decimal value of 01101 is 13 (between 1 and $(2^5 - 1) = 31$). Here, we add a constraint: the number of 1 bits of each combination must be odd. Thus, we actually have 2^{n-1} combinations available.

For each request, the VLR/SGSN chooses randomly a different number c $(1 \leq c \leq (2^n - 1))$ satisfying

the above constraint. According to its bit 1 positions $i_1, i_2, \cdots, i_m$, the VLR/SGSN retrieves the challenges $RN_{i_1}, RN_{i_2}, \cdots, RN_{i_m}$ from the vectors $AV[1..n]$. Then, it computes $RN_{VC} = RN_{i_1} \oplus RN_{i_2} \oplus \cdots \oplus RN_{i_m}$ and sends $VCREQ$ as *VC authentication request* to the MS:

$$VCREQ = \{c\}_{SK}, RN_{VC}.$$

Upon receipt of the message, the MS decrypts $\{c\}_{SK}$ and checks if combination c satisfies the constraint and if it is a new combination (all combinations sent from the VLR/SGSN are stored by the MS). If it does not, the MS rejects the authentication request. Otherwise, it accepts the request and retrieves the challenges $XRN_{i_1}, XRN_{i_2}, \cdots, XRN_{i_m}$ according to c from the vectors $MAV[1..n]$ and computes the result $XRN_{VC} = XRN_{i_1} \oplus XRN_{i_2} \oplus \cdots \oplus XRN_{i_m}$.

It checks if RN_{VC} and XRN_{VC} are equal. If they are not, the MS rejects the authentication request. Otherwise, the network authentication succeeds. The MS increases the value of the selected challenges by 1 and stores $XRN_{i_1} + 1, XRN_{i_2} + 1, \cdots, XRN_{i_m} + 1$ into the vector array $MAV[1..n]$ for next time authentication. Then, the MS computes $VCRES$ and sends it as *VC authentication response* to the VLR/SGSN:

$$VCRES = RES_{i_1} \oplus RES_{i_2} \oplus \cdots RES_{i_m} \oplus c.$$

After sending $VCRES$, the MS also generates the cipher key $CK = f3_{SK}(XRN_{VC})$, the integrity key $IK = f4_{SK}(XRN_{VC})$. After that, the MS stores the combination c into its database and updates the number of combinations used. If the number of combinations reaches 2^{n-1}, there is no more combinations available for future authentication. In this case, the MS discards the authentication vectors and the first procedure needs to be executed to request new authentication vectors for future authentication.

Upon receipt of the response, the VLR/SGSN computes

$$VCXRES = XRES_{i_1} \oplus XRES_{i_2} \oplus \cdots XRES_{i_m} \oplus c.$$

Then, it checks if $VCRES$ and $VCXRES$ satisfy the follow equation:

$$f2_{SK}(VCRES \oplus VCXRES) = R.$$

Here, we give an explanation. Since the number of 1 bits in c is odd, $VCRES$ can be written as $A \oplus R_y \oplus c$ and $VCXRES$ can be written as $A \oplus R_x \oplus c$. Thus, $VCRES \oplus VCXRES = R_x \oplus R_y$.

Note

that $R = f2_{SK}(R_x \oplus R_y)$ is computed in advance by the HE/HLR and stored only at the VLR/SGSN.

If they fail to satisfy this equation, the VLR/SGSN aborts the authentication. Otherwise, the MS authentication succeeds. The VLR/SGSN generates the cipher key and the integrity key similarly as the MS does: the cipher key $CK = f3_{SK}(RN_{VC})$ and the integrity key $IK = f4_{SK}(RN_{VC})$. Similarly, the VLR/SGSN also stores $RN_{i_1} + 1, RN_{i_2} + 1, \cdots, RN_{i_m} + 1$ into the vector array $AV[1..n]$ for next time authentication.

V. Security Analysis of VC-AKA

A. Enhanced Security

The proposed VC-AKA has enhanced security compared to UMTS AKA. In the first procedure, the distribution and establishment of authentication vectors, all the principals (the MS, the VLR/SGSN and the HE/HLR) contribute the generation of the session key SK and each RN_i with their own nonces. While in UMTS AKA, they are generated only by the HE/HLR. In the second procedure, to achieve mutual authentication between the MS and the VLR/SGSN, a randomly chosen combination c is sent to the MS. Only if c has not been used before can it be accepted by the MS. Thus, a replay attack is defended. The response from the MS is generated based on the contributions of all principals. Hence, mutual authentication is stronger. The cipher key CK and the integrity key IK are generated in the similar way.

VC-AKA does not employ sequence numbers to realize network authentication as in UMTS AKA. Instead, all three principals contribute the random numbers' generation and the network authentication is guaranteed through unused vector combinations. The only extra space needed for the VLR/SGSN and the MS is the storage for used combinations. In order to avoid large space for all 2^{n-1} combinations, the HE/HLR can tune n to a reasonable value (e.g. $n = 7$) or choose randomly only a part of combinations permitted for both the VLR/SGSN and the MS.

B. Against Malicious Threats

In UMTS security architecture [10], the VLR/SGSN could be in a foreign network and could be malicious. It is possible that the VLR/SGSN deliberately leaks Alice's information to Eve and tries to help Eve pass the authentication. Suppose that Eve has Alice's $AV[1..n]$. However, since $XRES_i$ is not equal to RES_i, Eve cannot compute the correct $VCRES$ without the knowledge of R_y which is sent to Alice's mobile station secretly.

For each time successful authentication, both the VLR/SGSN and the MS update the challenges by increasing their values by 1. Thus, the challenges in the two arrays are always fresh. Even if an adversary intercepts several used combinations and RN_{VC} values, he can do nothing and cannot concoct a valid RN_{VC} or XRN_{VC} to impersonate a legitimate entity.

The session key SK helps the VLR/SGSN and the MS to generate the cipher key CK and the integrity key IK in the similar way as in [15]. Even if Eve gets Alice's SK in the help of the malicious VLR/SGSN, however, Eve can do nothing. Note that $R = f2_{SK}(R_x \oplus R_y)$ is computed by the HE/HLR in advance; the VLR/SGSN can use R for verification only and cannot obtain the knowledge of R_y due to the one-way hash function; the MS has no knowledge of R and R_x, either.

C. Against Redirection Attacks

In [14], the authors showed a possible redirection attack in UMTS AKA because the authentication vector can be used by any serving network. In VC-AKA, when the MS roams from the HE/HLR to a VLR/SGSN, it receives the *user data request* message and replies a *user data response* message which includes the $MAC_M = f1_K(N_M \ N_V \ V \ H)$. When the HE/HLR receives the *authentication data request* message, it can check if the MS is really in the coverage of the supposed VLR/SGSN. If it is not, the HE/HLR refuses the connection request. While the MS receives the authentication token $AUTN$, he can check if they are really sent by the supposed HE/HLR and VLR/SGSN through MAC since all three principals contribute to the generation of MAC. Similarly, for each mutual authentication between the MS and the VLR/SGSN, all principals also contribute to the generation of challenge/response message. Hence, VC-AKA can defend the redirection attack.

D. Network Corruption Impact

We also examine the impact of network corruption discussed in [14]. Suppose that a VLR/SGSN is corrupted and the adversary can eavesdrop any message received or sent by VLR/SGSN. There are two possible scenarios. First, the MS has no authentication vectors left and needs to execute the first procedure to request vectors from its HE/HLR. However, in this scenario, the adversary cannot forge an authentication data request message because this message should comprise a message authentication code from the MS.

Second, the MS has unused authentication vectors when the VLR/SGSN is corrupted. The adversary can concoct a challenge and force the MS to response. However, the adversary can only use the left combinations and can do nothing when all combinations are used up. In this case, the first procedure

should be executed to obtain new authentication vectors while it is impossible as described in the first scenario. Hence, network corruption can only influence the MS who has unused vector combinations. While in UMTS AKA, it could influence all the legitimate users.

VI. Comparison with Related Work

During the first procedure, we need more protocol rounds to establish the authentication vectors in VLR/SGSN and the MS. This would bring more traffic than UMTS AKA. However, it is worth as these vectors can be used much more times than UMTS AKA in the second procedure. This is realized through the introduction of vector combination. Consequently, for an array of n vectors, authentication can reach up to 2^{n-1} times instead of only n times in UMTS AKA. Take an example, $n = 7$, and authentication can happen up to 64 times instead of only 7 times. This decreases exponentially the traffic needed to generate authentication vectors for the HE/HLR which has numerous subscribers.

We compare VC-AKA to other proposals. Recently, an analytic model [17] is proposed to investigate the impact of the size of the authentication vector array in order to minimize the cost. Another work [18] proposes a dynamic length of authentication vector array based on prediction of the mobile user's residence time in the VLR/SGSN. Consequently, it is able to reduce the network traffic and avoid the bottleneck at AuC. These works are different from VC-AKA because they do not change the original UMTS AKA protocol and tries to find a best size of the array through traffic analysis.

In [14], an adaptive AKA (AP-AKA) is proposed without sequence numbers. Instead, the MS keeps a list of unused vectors' indexes to verify if an authentication vector is really sent from the serving network and was not used before. This helps the mobile user to defend replay attacks. The protocol also differentiates if the MS stays in his home network or roams to a serving network. In each scenario, the execution of protocol is different. To some extent, this improves the efficiency since the protocol in the home network needs less runs than that in the serving network. As in UMTS AKA, the authentication vectors can also be used only n times.

In [15], the authors employ two techniques: a one-time password/hash chaining technique [21] and keyed-Hash Message Authentication Code (HMAC) [22] instead of the sequence number to establish mutual authentication. For convenience, we call this HH-AKA. In HH-AKA, when a mobile user roams to a VLR, he generates a hash chain and sends to his HE/HLR. After successful verification of the hash chain, the HE/HLR sends it to the VLR. The VLR then generates another hash chain and sends to the MS. Thus, each MS and VLR pair has two distinct hash chains for successive mutual authentication.

This mechanism, however, introduces another synchronization: the authentication times between the MS and the VLR. Once a mismatch of times occurs due to a network failure, it is difficult for the two parties to synchronize. Compared to HH-AKA, the combination check in VC-AKA is more flexible: if a combination is found to be used, it does not influence future authentication. As in UMTS AKA and [14], authentication can happen only up to the length of the hash chain.

As in UMTS AKA, the redirection attack in HH-AKA is also possible as discussed in [14]. An adversary can impersonate a base station and entice the mobile user to camp on the radio channels of the false base station. Since there is no serving network identity indication in the registration message, the home network cannot differentiate whether the message is from the mobile user directly or relayed by an adversary. Consequently, the authentication would be successful and the adversary could redirect the traffic to an unintended network.

Another drawback in HH-AKA is the employment of timestamps for message freshness while the MS sends his hash chain to the HE/HLR. However, to guarantee freshness by timestamps, both MS and HE/HLR must keep local clocks periodically synchronized by a reliable source of time in a secure manner. Between synchronization with the reliable time source, local clocks may drift [23]. Both entities must allow a time window for timestamps to compensate for local clock drift and the fact that messages take time to cross a network. Moreover, the mobile user might have his own "personalized clock" in his real life, for example, always 3 minutes in advance for not missing the bus. Thus, it is not reasonable to require the mobile user to keep an exact clock as the network.

The timestamps problem exists also in X-AKA proposed by [16]. In X-AKA, the mobile user sends his message authentication code based on the timestamp, and the temporary key generated is also a function of the timestamp. Because of the disadvantage of timestamp, the message receiver could reject the message. As in UMTS AKA and in [15], there is also a potential redirection attack as discussed in [14].

VII. CONCLUSIONS

This paper first gave an overview of UMTS AKA and showed its weaknesses, especially the adoption of sequence numbers and the traffic bottleneck in the HE/HLR. The synchronization of sequence numbers has been criticized a lot and many improvements have been proposed. In this paper, in order to eliminate these weaknesses and enhance the security, we abandon completely the adoption sequence numbers and propose an enhanced AKA based on authentication vector combination. The main advantage of VC-AKA is that it liberates the HE/HLR from the bottleneck of authentication vectors generation.

The security analysis showed that VC-AKA can defend against redirection attacks and active attacks in corrupted networks. Also, VC-AKA provides enhanced security on session key generation and mutual authentication. Through comparison with UMTS AKA and its improvements in literature, we believe that VC-AKA is more efficient and secure.

REFERENCES

[1] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Communications Magazine*, vol. 31, pp. 92–100, 1993.

[2] L. Salgarelli, M. Buddhikot, J. Garay, S. Patel, and S. Miller, "Efficient authentication and key distribution in wireless IP networks," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 52–61, Dec. 2003.

[3] G. M. Koien and T. Haslestad, "Security aspects of 3G-WLAN interworking," *IEEE Communications Magazine*, vol. 41, no. 11, pp. 82–88, Nov. 2003.

[4] J. Al-Muhtadi, D. Mickunas, and R. Campbell, "A lightweight reconfigurable security mechanism for 3G/4G mobile devices," *IEEE Wireless Communications*, vol. 9, no. 2, pp. 60–65, Apr. 2002.

[5] Y.-B. Lin, M.-F. Chang, M.-T. Hsu, and L.-Y. Wu, "One-pass GPRS and IMS authentication procedure for UMTS," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1233–1239, Jun. 2005.

[6] H. Chen, M. Zivkovic, and D.-J. Plas, "Transparent end-user authentication across heterogeneous wireless networks," in *Vehicular Technology Conference, VTC 2003-Fall*, vol. 3, Oct. 6–9, 2003, pp. 2088–2092.

[7] Y.-B. Lin, M.-F. Chen, and H.-H. Rao, "Potential fraudulent usage in mobile telecommunications networks," *IEEE IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 123–131, Apr.-Jun. 2002.

[8] F. Fitzek, M. Munari, V. Pastesini, S. Rossi, and L. Badia, "Security and authentication concepts for UMTS/WLAN convergence," in *Vehicular Technology Conference, VTC 2003-Fall*, vol. 4, Oct. 6–9, 2003, pp. 2343–2347.

[9] S.-H. Lim, H. S. Roh, D.-I. Kim, and S. ho Lee, "Authentication architecture for interworking between universal mobile telecommunications systems (UMTS) and high-speed portable internet (HPI) system," in *Vehicular Technology Conference, VTC 2005-Fall*, vol. 2, Sep. 25–28, 2005, pp. 774–778.

[10] *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 7)*, 3GPP Std. TS 33.102 V7.0.0, Dec. 2005.

[11] K. Boman, G. Horn, P. Howard, and V. Niemi, "UMTS security," *Electronics & Communication Engineering Journal*, pp. 191–204, Oct. 2002.

[12] G. Rose and G. Koien, "Access security in CDMA2000, including a comparison with UMTS access security," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 19–25, Feb. 2004.

[13] G. M. Koien, "An introduction to access security in UMTS," *IEEE Wireless Communications*, vol. 1, no. 1, pp. 8–18, Feb. 2004.

[14] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE transactions on wireless communications*, vol. 4, no. 2, pp. 734–742, Mar. 2005.

[15] L. Harn and W. Hsin, "On the security of wireless network access with enhancements," in *Proceedings of the 2003 ACM workshop on Wireless security*, San Diego, USA, Sep. 19 2003, pp. 88–95.

[16] C. Huang and J. Li, "Authentication and key agreement protocol for UMTS with low bandwidth consumption," in *19th International Conference on Advanced Information Networking and Applications (AINA 2005)*. Taipei, Taiwan: IEEE Computer Society, Mar. 28–30, 2005, pp. 392–397.

[17] Y.-B. Lin and Y.-K. Chen, "Reducing authentication signaling traffic in third-generation mobile network," *IEEE Transactions on Wireless Communications*, vol. 2, no. 3, pp. 493–501, May 2003.

[18] J. Al-Saraireh, S. Yousef, and M. A. Nabhan, "Analysis and enhancement of authentication algorithms in mobile networks," *World Wireless Congress*, vol. 2006, pp. 225 – 230, 2006.

[19] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, Feb. 1990.

[20] C. Boyd and W. Mao, "On a limitation of BAN logic," in *In Advances in Cryptology: Eurocrypt '93*. Springer-Verlag, 1993, pp. 240–247.

[21] L. Lamport, "Password authentication with insecure communication," *Communications ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[22] H. Krawczyk, M. Bellare, and R. Canetti, "Keyed-hashing for message authentication," *Internet Engineering Task Force, Request for Comments (RFC) 2104*, Feb. 1997.

[23] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.