Novel Authentication Algorithm – Public Key Based Cryptography in Mobile Phone Systems

Győző Gódor,[†] and Sándor Imre Dr.[††],

[†,††] Department of Telecommunications, Budapest University of Technology and Economics, Magyar tudósok körútja 2. H-1117, Budapest, Hungary

Summary
Nowadays in the field of telecommunications and computer sciences the most important part of development is security. In UMTS, the weaknesses of GSM solved but some problems in the security area are revealed, e.g. the IMSI is plain on the air interface. We introduce our novel authentication algorithm which solves this fault. The GSZV algorithm is based on public key infrastructure which is used in the computer network security. Because of the convergence of various telecommunication systems it is worth considering the utilization of public key cryptography in mobile phone systems. Our method is an example of how this kind of cryptography can be used.

Key words:
Authentication, public-key, cryptography, mobile security

Introduction

Nowadays in the area of telecommunications and computer sciences the mobility is very substantial. Users demand the secure, authoritative and reliable communication. Hence, many ideas are evolved to improve the security of the mobile networks [1].

In the GSM system one of the most criticized security issue is that the sensitive control data – keys used for air interface ciphering – are sent to different networks without ciphering. In the UMTS, the security weaknesses of GSM are perceived and improved, however, some problems in the security field are revealed, e.g. the IMSI is not encrypted on the air interface [2]. If the IMSI is plain, an unauthorized person can abuse it or can be tracking the user [6].

With our novel authentication algorithm the above-mentioned problem can be solved. The GSZV authentication algorithm is based on public key infrastructure which is used in the computer network security. Because of the convergence of various telecommunication systems it is worth considering the utilization of public key cryptography in mobile phone systems. Therefore we developed an algorithm that helps the user and the network to authenticate each other without having to send unencrypted information.

This paper is organized as follows: In Section 2 an overview of the UMTS security is given. In Section 3 we introduce our novel authentication algorithm, called GSZV authentication algorithm, whose name derives from the abbreviation of the developers' names. In Section 4 we present how our algorithm can be protected against different kinds of attacks. Comparing to the UMTS AKA the advantages and disadvantages of our algorithm are explained in Section 5. We implemented the authentication algorithm in OMNeT++ and in Section 6 we introduce several screenshots about the simulation. Finally, in Section 7 we draw the conclusions.

2. Overview of the UMTS security

The UMTS security architecture is defined by ETSI. For 3GPP, the access security architecture is specified in 3G TS 33.102 (Release 1999) [2] [7] which contains the following major features:

- A 128-bit encryption algorithm (the KASUMI algorithm [3]).
- A mutual authentication mechanism between the user and the network.
- Enhanced cryptographic algorithms for Authentication and Key Agreement (AKA).
- The MILENAGE algorithm set (f1, f1*, f2, f3, f4, f5 and f5*) [4].
- A new integrity check mechanism for signaling messages.

2.1 Terminology

- SN (Serving Network), it contains:
  - VLR (Visitor Location Register)
  - SGSN (Serving GPRS Support Node)
- HN (Home Network)
  - HLR (Home Location Register)
- AuC (Authentication Center which belongs to the HLR)

Manuscript received February 25, 2006.
Manuscript revised February 28, 2006.

- K master key (Subscriber Authentication Key with a length of 128 bits)
- UE/USIM (User Equipment / UMTS SIM)
  - K master key
- RNS (Radio Network System)
  - Node B (UMTS base station)
  - RNC (Radio Network Controller): control equipment for Node B
- RAND (random number for the challenge-and-response mechanism)
- (X)RES (Expected User Response)
- CK (Encryption Key)
- IK (Integrity Key)
- AUTN (Authentication Token for Network Authentication)
- {RAND, XRES, CK, IK, AUTN}: security quintet
- SQNms (Sequence Number Information at User)
- SQNhe (Sequence Number Information at Home System)
- IMSI (International Mobile Subscriber Identity)
- TMSI (Temporary Mobile Subscriber Identity)

The terminology of the UMTS system is presented in the [7] and [8] in detail.

2.2 UMTS Authentication and Key Agreement

The UMTS authentication is based on the symmetric key cryptography [9]. The shared secret is the K master key which is implemented in the USIM (Universal Subscriber Identity Module) and the Authentication Center (AuC) of the Home Network (HN). The length of master key is 128 bits [11].

According to the GSM in the UMTS standard the mutual authentication is specified which means the user and the serving network authenticate each other. While in the GSM system the encryption is used on the radio interface, in the UMTS it is applied on the backbone even though the risk of an attack is very low here [8].

Towards the protection of the subscriber the IMSI (International Mobile Subscriber Identity) is passed over the radio interface only once, then the granted TMSI (Temporary Mobile Subscriber Identity) will be used for authentication. However, the IMSI is plain-text in this operation therefore the user will be traceable, one's position or habitation will be determinable which have the chance of misuse.

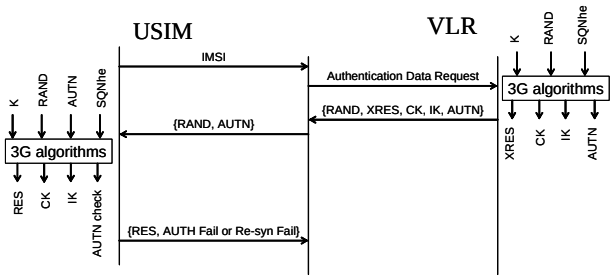

Fig. 1 The authentication algorithm of UMTS.

The authentication method can be seen in Fig. 1. Briefly the authentication mechanism functions in the following way:

The authentication procedure can be started after the user is identified in the serving network. The identification occurs when the identity of the user, i.e. permanent identity IMSI or temporary identity TMSI, has been transmitted to VLR or SGSN. Then VLR or SGSN sends an authentication data request to the Authentication Centre (AuC) in the home network (see Fig 1.) [11].

The AuC stores the master keys of users and based on the IMSI the AuC is able to generate authentication vectors for the user. The authentication vector contains five parameters (RAND, XRES, CK, IK, AUTN), which is called "5-touple". This is done by means of the authentication algorithms and the users' private secret key K (see Fig 2.). K is only found in the AuC and on the USIM. The generated vectors are sent back to the VLR/SGSN in the authentication data response [7].

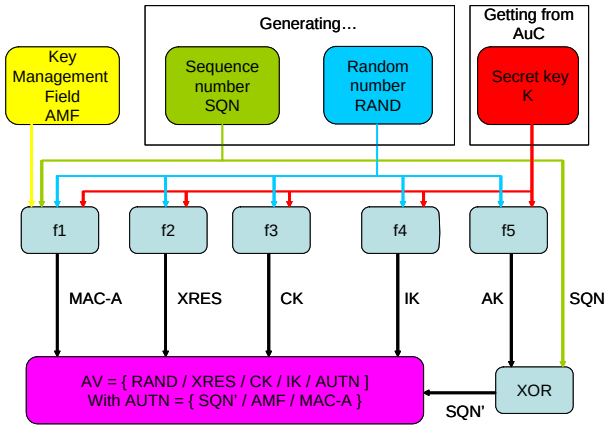

Fig. 2 Authentication Vector Generation on the AuC side

In the serving network, one authentication vector is needed for each authentication instance, i.e. for each run of the authentication procedure. This means the signaling between SN and the AuC is not needed for every authentication event and it can in principle be done

independently of the user actions after the initial registration. Indeed, the VLR/SGSN may fetch new authentication vectors from AuC well before the number of stored vectors runs out.

The VLR/SGSN challenges the user/USIM by sending a user authentication request to the terminal. This message contains two parameters from the authentication vector (RAND and an Authentication token (AUTN)) [12].

The user/USIM processes the AUTN. The AUTN contains a sequence number (SEQ), a message authentication code (MAC-A) and an authentication management field (AMF). With the aid of the private secret key K, the user is able to verify that the received challenge data could only have been generated by someone who has the same secret key K. The user/USIM will also verify that the AV has not expired by checking the sequence number (SEQ). Provided that the network can be authenticated and that the AV is still valid (fresh), the user/USIM proceeds to generate the confidentiality key (CK), the integrity key (IK) and the response (RES).

The user/USIM responds with the RES to the network.

The VLR/SGSN verifies that response is correct by comparing the expected response (XRES) which is part of the AV it received from the AuC with the response (RES) received from the user/USIM.

The generation of keys and the computing of the Response are derived by public algorithms:

- MAC code: f1 algorithm
- RES: f2 algorithm
- CK, IK: f3-f5 algorithms

The authentication mechanism of UMTS system is presented in [8] in detail.

3. GSZV Authentication Algorithm

In this Section we introduce our novel, public key infrastructure-based authentication algorithm. However, firstly we explain our terminology.

3.1. Terminology

- $E\{M\}$ means: data M is encrypted by the public key of X
- $E_{HLR}\{M\}$ means: data M is encrypted by the public key of the HLR
- $E_{VLR}\{M\}$ means: data M is encrypted by the public key of the VLR
- $E_X\langle\{A,B\}C_Y\rangle$ means: A and B are signed by the certification key of Y entity and they are encrypted by the public key of X. The certification key is a secret signing key which belongs to the given entity. Inside the brace the different data are separated by commas.
- $\{M\}C_Y$ means: M signed by the signing key of Y
- Two different keys are used in the authentication algorithm. These keys are the certification key and the encryption key. The VLR and the HLR possess two different key-pairs.

- USIM (Universal Subscriber Identity Module)
  - K_P, K_S (public and secret key of the subscriber)
  - IMSI
  - Certificate of the HLR: $\{E_{HLR}\langle IMSI, K_P\rangle\}C_{HLR}$, which means that the provider signs that subscriber is theirs (CERT).
  - SQN (sequence number), it must be an integer number and its initial value is zero
- HLR (Home Location Register)
  - $K_{P_{HLR}}$, $K_{S_{HLR}}$ (public and secret encryption key of the User's Service Provider)
  - $S_{P_{HLR}}$, $S_{S_{HLR}}$ (public and secret certification key of the User's Service Provider)
- VLR (Visitor Location Register)
  - $K_{P_{VLR}}$, $K_{S_{VLR}}$ (public and secret encryption key of the Visited Network Operator)
  - $S_{P_{VLR}}$, $S_{S_{VLR}}$ (public and secret certification key of the Visited Network Operator)

3.2. GSZV Algorithm Method

The GSZV authentication algorithm is based on public key infrastructure. The mutual authentication, the integrity and non-repudiation are guaranteed by using digital signatures and certificates [10]. Applying challenge and response technique the replay attacks can be efficiently prevented.
Our authentication algorithm can be seen in Fig. 3. The method is the following:

When the user enters a VN the USIM receives $K_{P_{VLR}}$ which is broadcasted by the VLR. The USIM sends the CERT and the sequence number (SQN) encrypted by the public key of the HLR ($K_{P_{HLR}}$). On all occasions when a message is sent by the terminal the SQN value is incremented and stored on the USIM. SQN is encrypted and only the HLR can decrypt it. The whole message is encrypted by the public key of VLR so the first message is the following: $E_{VLR}\langle CERT, E_{HLR}\langle SQN\rangle\rangle$.

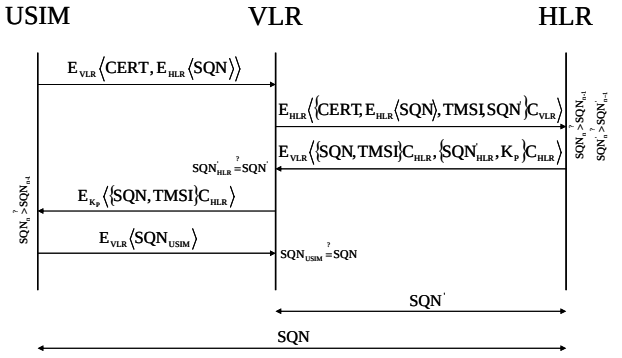

Fig. 3 GSZV authentication algorithm

When the VLR receives the previous message, it decrypts it by using $K_{S_{VLR}}$. The certificate of the HLR is retrieved by the VLR but other information (IMSI, SQN) is not accessible. Supposing that each service provider has roaming contracts, they know each other's public certification keys. Thus the VLR can check the authenticity of the signature and learns which service provider issued the certificate to the user.

In the next step the VLR generates a TMSI and another sequence number (SQN'). This SQN' can be ensured that every message is unique between the VLR and the HLR. The two different SQN can be seen at the bottom of Fig. 3. These SQN numbers protect against replay attacks. The VLR signs the CERT, the $E_{HLR}\langle SQN\rangle$, the TMSI and the SQN' and encrypts it by $K_{P_{HLR}}$. After that the VLR forwards this message to the HLR, and the SQN' value is incremented by one in its database. The second message is the following: $E_{HLR}\langle\{CERT, E_{HLR}\langle SQN\rangle, TMSI, SQN'\}C_{VLR}\rangle$.

The HLR receives this message and decrypts it by $K_{S_{HLR}}$. By using the public certification key of VLR the HLR can verify that the VLR is a legitimate network element. The HLR encrypts the SQN value which is sent by the USIM and make comparison between the stored SQN value and the received one. This mechanism supplies the protection of the replay attacks between the UE and the HLR. Then the HLR extracts the SQN' and can be decided if the message sent by the VLR is fresh or not. In our model a message is fresh when the received SQN' value is bigger than the stored one. It is essential to protect the messages against replay attacks between the VLR and the HLR. If the message is not fresh the HLR knows that the VLR is not a legitimate network element and discards the message. The IMSI can be picked up from the CERT with the help of which HLR can decide that the user is its own subscriber or not. If the IMSI is in the database then the user is authenticated. The HLR stores the TMSI along with the IMSI using for subsequent registration.

The replay message of the HLR is the following: $E_{VLR}\langle\{SQN, TMSI\}C_{HLR}, \{SQN'_{HLR}, K_P\}C_{HLR}\rangle$. The public key of USIM (K_P) is recovered from the CERT. Then the $\{SQN, TMSI\}$ is signed by the HLR and the SQN' value (SQN'_{HLR}), which was received from the VLR and the K_P, are signed, too. So this challenge and response technique protects against replay attacks. The previous two messages are encrypted by $K_{P_{VLR}}$ and sent to VLR.

When the VLR receives the third message, decrypts it by $K_{S_{VLR}}$ and checks the authenticity of the signature. Now the VLR is able to compare the two SQN' values (SQN' and SQN'_{HLR}). If these are equal the VLR knows that the user is really authenticated and the HLR is really the network element which received the last message transmitted by the VLR in the previous step.

If the authentication process is passed the $E_{K_P}\langle\{SQN, TMSI\}C_{HLR}\rangle$ is sent to the USIM through an authenticated and secure channel opened by the VLR. The user's public key has been retrieved after processing the second message only by the HLR. Being bound to the target is valid as the VLR can get to know the user's public key only after the VLR and the HLR proved their identity. The IMSI is known exclusively by the HLR throughout the whole authentication algorithm.

USIM receives the message, decrypts it by K_S and checks the digital signature of the HLR then extracts the TMSI and the SQN. USIM decides whether the message is replayed or not by using the SQN value. If it is not a replayed message the USIM will return the SQN to the VLR (SQN_{USIM} is encrypted by $K_{P_{VLR}}$) in order to prove one's identity. This challenge and response technique is the same as the one used between the VLR and the HLR. The VLR receives the $E_{VLR}\langle SQN_{USIM}\rangle$ message and compares the original and the received SQN. If these match, the user is authenticated and the connection is established. If SQN is not equal or the message is not received from the VLR in a given time interval, the user is disconnected and the authentication mechanism is started again.

4. Protection against different attacks

In this section we introduce the resistance of GSZV authentication algorithm against different attacks.

4.1. Replay Attack

Technical definition: an attack in which a valid data transmission is maliciously or fraudulently repeated, either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack.

Using the two different sequence numbers, we can ensure that each message is unique among the VLR, the USIM and the HLR. SQN defends between the USIM and the HLR, $SQN^{'}$ defends between the VLR and the HLR against a replay attack.

4.1.1 USIM → Attacker → VLR

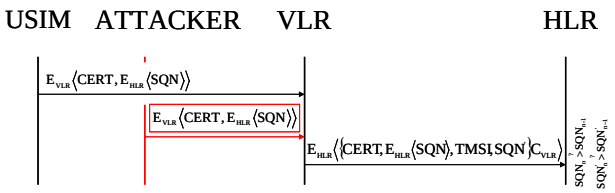

Fig. 4 Replay attack (USIM → Attacker → VLR)

When the USIM sends the $E_{VLR}\langle CERT, E_{HLR}\langle SQN\rangle\rangle$ message (see Fig. 4.) to the VLR, the SQN value is incremented by one and stored in a database on the USIM. An attacker intercepts the data and retransmits it to the VLR without modifying its content. The VLR receives it, thus the following message is forwarded: $E_{HLR}\langle\{CERT, E_{HLR}\langle SQN\rangle, TMSI, SQN^{'}\}C_{VLR}\rangle$.

When the HLR retrieves the SQN value and compares it to the SQN value which is in its database the inequality relation $SQN_n > SQN_{n-1}$ does not exist, therefore the HLR knows that the message from the UE is replayed and discards it.

If the SQN were not in our algorithm, the attacker would always intercept the same message between the USIM and the VLR – on condition that we talk about the same USIM and the same VLR – and he could be authenticated himself towards the HLR. However, the attacker would not be able to decrypt the fourth message ($E_{K_P}\langle\{SQN, TMSI\}C_{HLR}\rangle$ see Fig. 3.) because he does not know the secret key of the USIM.

Therefore an attacker could only personalize as a legitimate network element to deceive the HLR and could see the replay messages. By this information he may decrypt the messages using statistical methods.

4.1.2 VLR → Attacker → HLR

The following event is when the attacker receives the message which was sent from VLR to HLR and later transmits it to the HLR (see Fig. 5.).

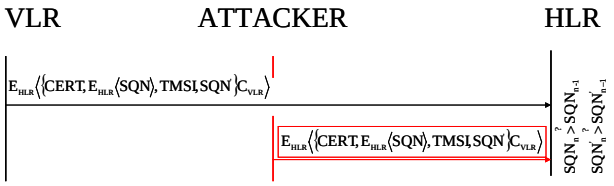

Fig. 5 Replay attack (VLR → Attacker → HLR)

The $E_{HLR}\langle\{CERT, E_{HLR}\langle SQN\rangle, TMSI, SQN^{'}\}C_{VLR}\rangle$ message is received by HLR and is decrypted by $K_{S_{HLR}}$. The HLR can check the authenticity of the digital signature of VLR and, similarly in Section 4.1.1, the HLR decides that the retrieved $SQN^{'}$ value is bigger or not than the $SQN^{'}$ value stored in its own database ($SQN_n^{'} \stackrel{?}{>} SQN_{n-1}^{'}$). If the inequality relation does not exist, the message was a replayed one, thus the HLR can recognize the attack.

4.1.3 VLR ← Attacker ← HLR

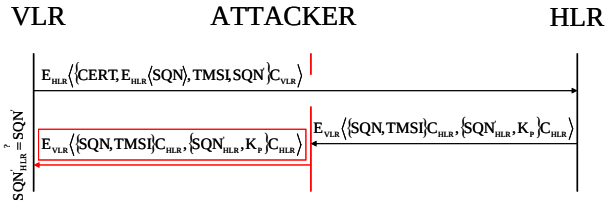

Fig. 6 Replay attack (VLR ← Attacker ← HLR)

In this instance the attacker intercepts the message transmitted by HLR and some time later forwards it to the VLR without modifying its content (see Fig. 6.). The VLR retrieves the $SQN_{HLR}^{'}$ value from $E_{VLR}\langle\{SQN, TMSI\}C_{HLR}, \{SQN_{HLR}^{'}, K_P\}C_{HLR}\rangle$ and compares the transmitted and the received $SQN^{'}$. If these match, the VLR will be authenticated, otherwise the VLR knows that the message received from the HLR was a retransmitted message.

4.1.4 USIM ← Attacker ← VLR

In this case the attacker intercepts and subsequently forwards the message which was sent by VLR (see Fig. 7.) to USIM.

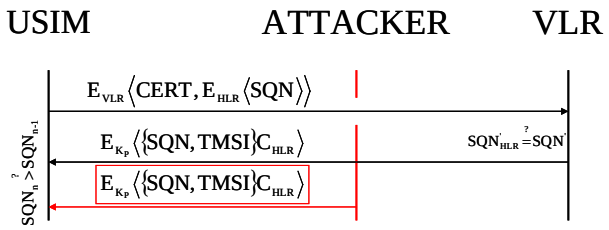

Fig. 7 Replay attack (USIM ← Attacker ← VLR)

The UE compares the two SQN values, one of them is found in the database of USIM, the other can be get from the received message. The inequality $SQN_n > SQN_{n-1}$ does not exist, since when the message was repeated the SQN value had already been incremented at least once on the USIM side.

4.1.5. USIM → Attacker → VLR

Even in the last step of the authentication an attack may occur as it can be seen in Fig. 8.

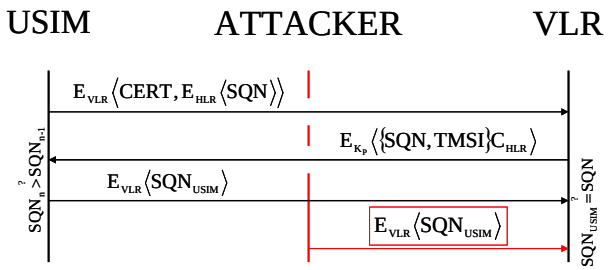

Fig. 8 Replay attack (USIM → Attacker → VLR)

The attacker intercepts the response message of the USIM for the challenge of VLR and another time relays it to the VLR without modifying its content. By using $K_{S_{VLR}}$ the VLR retrieves the SQN_{USIM} value and compares it with the SQN value stored in its own database. The attack is an unsuccessful attempt because if the VLR did not receive any new message between the intercepting and the replaying then the VLR discards the same messages automatically. If new message was received, the VLR increments the value of the SQN.

4.2. Man-in-the-middle

Technical definition: a form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication association.

The digital signatures protect against data-modifying because the secret certification keys are only known by the VLR and the HLR. The attacker is not able to retrieve any information from the intercepted data because of the use of digital signatures, certifications and the encryption.

4.3. Personalization

In this instance we analyze that an attacker personalizes the VLR as it can be seen in Fig. 9.

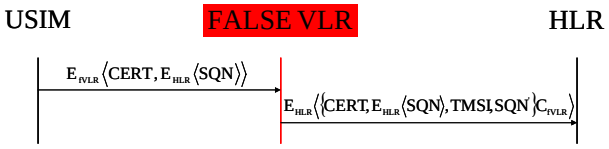

Fig. 9 Personalization

The personalized VLR broadcasts a false public key for encryption. By using the false public encryption key the USIM generates the following message: $E_{fVLR}\langle CERT, E_{HLR}\langle SQN\rangle\rangle$. The attacker is able to retrieve the CERT and the $E_{HLR}\langle SQN\rangle$ but it cannot decrypt them. In the next step the attacker sends the $E_{HLR}\langle\{CERT, E_{HLR}\langle SQN\rangle, TMSI, SQN'\}C_{fVLR}\rangle$ message in which the data $\{CERT, E_{HLR}\langle SQN\rangle, TMSI, SQN'\}$ is signed by the false digital signature (C_{fVLR}). Since the false VLR does not know the secret certification key of any legitimate VLR thus he is not able to generate an authentic digital signature. When the HLR receives the signed message which is encrypted by the public key of the HLR, checks the authenticity of the signature and knows that the VLR is not a legitimate network element.

Similarly to this example an attacker is not able to authenticate himself personalized by the HLR or the USIM.

5. Comparing the GSZV Authentication Algorithm with the UMTS AKA

In this section we compare the GSZV authentication algorithm with the UMTS authentication algorithm from different points of view. We introduce that the GSZV authentication algorithm fulfils higher security requirements, has simpler set-up and protects more efficiently against various attacks.

5.1. Cryptography

The GSZV authentication algorithm is based on the public key cryptography, while the UMTS AKA is based on the symmetric key cryptography which is a fundamental difference. The usage of symmetric key cryptography is disadvantageous for several reasons: the keys are shared

and if somebody obtains the key of an entity then he will be able to decrypt all the messages encrypted by the key. In the GSZV algorithm the secret keys for encryption and signature are not shared and the private keys are stored only by the corresponding entities.

In our novel authentication method the computation complexity is higher but communication complexity of the key management is simpler than in the UMTS environment. The VLR does not need to store the public keys of the users, it is enough to store only the certification and encryption key-pairs of the HLR and the VLR. But we are able to create secure HLR and it has a sufficiency of capacity.

5.2. Security

Nowadays the public key cryptography is used in the computer network security and the Internet communications for authentication and encryption. This mechanism is a secure way to protect the identity of the user.

In our solution the IMSI and the public key of the user are stored on the USIM in an encrypted form. This information is encrypted by the public encryption key of the HLR therefore only an authenticated VLR is able to learn the identity of the user.

In the UMTS the IMSI is sent in the plain on the air interface thus an attacker can obtain the identity of the user by personalizing the VLR or simply by intercepting the message. Additionally in the GSZV algorithm the principles of non-repudiation, authentication and message integrity are fulfilled, while for example in the first step of the UMTS AKA these principles are not fulfilled between the USIM and the VLR.

5.3. Flooding

In the UMTS it is possible to flood the system by resending false IMSI messages because the IMSI is not encrypted from the USIM to the VLR. The digital signatures, the public key cryptography and the two different sequence numbers protect together against the flood of modified messages or the replay of messages sent by any element of the network.

6. Simulation of the GSZV Authentication Algorithm in OMNeT++

Our authentication algorithm was implemented in OMNeT++, therefore firstly, we give a brief overview about OMNeT++, after that we introduce the simulation.

6.1. Overview of the OMNeT++

OMNeT++ [13] is an object-oriented modular discrete event network simulator. The simulator is an open-source product, it is free to download for academic and non-profit use.

The simulator can be used for traffic modeling of telecommunication networks, protocol modeling, modeling multiprocessors and other distributed hardware systems or modeling any other system where the discrete event approach is suitable.

An OMNeT++ model consists of hierarchically nested modules. The depth of module nesting is not limited, which allows the user to reflect the logical structure of the actual system in the model structure. Modules communicate through message passing. Messages can contain arbitrarily complex data structures. Modules can send messages either directly to their destination or along a predefined path, through gates and connections.

Developers have to code the behaviour of modules, which are the lowest level of the module hierarchy. These modules are programmed in C++. Modules can have their own parameters with the help of which the behaviour of modules can be customized and the topology of modules can be parameterized.

OMNeT++ simulations have various user interfaces which have different functions: debugging, demonstration and batch execution.

The whole simulator – user interfaces, tools, etc. – is written in C++ which guarantees the portability. Therefore the simulation works in Windows and Linux operating systems using various C++ compilers.

The OMNeT++ has a special description language, the Network Description Language (NED), by which the parameters of modules, the connections between modules and the gates can be defined. The modules and the whole network can be designed graphically, too. The NED files can be compiled to C++, thus the whole program can be portable.

OMNeT++ also supports parallel distributed simulation. OMNeT++ can use several mechanisms for communication between partitions of a parallel distributed simulation, for example MPI or named pipes. The parallel simulation algorithm can easily be extended or new ones plugged in. OMNeT++ can even be used for classroom presentation of parallel simulation algorithms, because simulations can be run in parallel even under the GUI which provides detailed feedback on what is going on.

6.2. The simulation of the GSZV authentication algorithm

We implemented our above-mentioned authentication algorithm (Section 3.). The graphical interface of the simulation can be seen in Fig. 10.

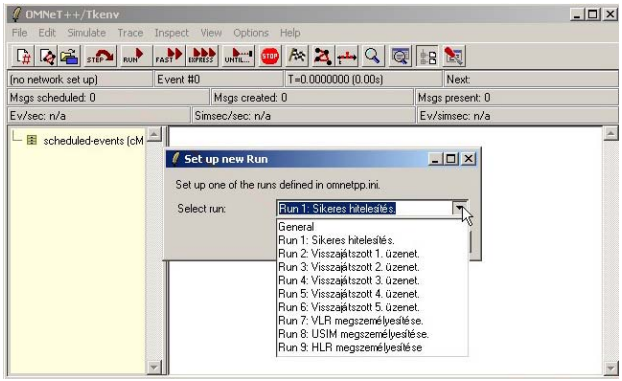

Fig. 10 The graphical interface of the simulation

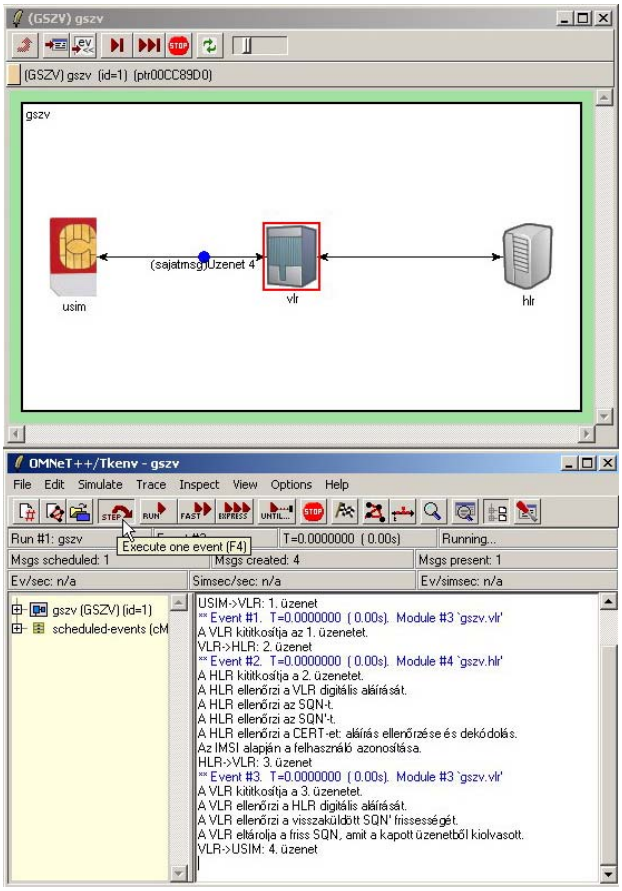

Fig. 11 The simulation of the GSZV authentication algorithm

The whole authentication algorithm and each attack method which we introduced in Section 4 are coded so the program has 9 different runs. The beginning of the simulation we can choose one of them as it can be seen in Fig. 10.

Fig. 11 shows the simulation in action.

We examined each attack mechanism by the simulator and we discovered that our algorithm is resisted all of them as it mentioned earlier. So the theoretical thoughts correspond with the results of the simulation therefore we demonstrated that the GSZV authentication algorithm stands one's ground in mobile environment.

7. Conclusions and Future Work

In this contribution we have introduced our novel authentication algorithm called GSZV authentication algorithm which is based on the public key infrastructure. It fulfils higher security requirements than the UMTS authentication scheme. It guarantees a secure and confidential communication between the users and the network.

In our solution all information, including the IMSI of the user is encrypted on the air interface which is needed since if the IMSI gets known an attacker misuse it. We presented the efficiency of GSZV algorithm and compared with the UMTS AKA. Our algorithm protects against different kinds of attacks using digital signatures, certifications and two different sequence numbers.

In the OMNeT++ simulation we introduced our algorithm in action. As compared with the theoretical and the simulation results we proved that the GSZV authentication algorithm stands one's ground in mobile environment.

Our future works are integrating a key-creation and distribution algorithm into our novel solution and creating a test implementation to analyze this in real circumstances.

Acknowledgments

This work is supported by OTKA F042590.

References

[1] R. Safavi-Naini, W. Susilo, G. Taban, "Towards securing 3G mobile phones", Proceedings of Ninth IEEE International Conference on Networks, 10-12 Oct. 2001.

[2] 3GPP, 3G TS 33.102 "Security architecture"

[3] 3GPP, 3G TS 35.201 "Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification

[4] 3GPP, 3G TS 35.201 "Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3,f4,f5 and f5*; Document 1: General"

[5] L.I. Millett, S.H. Holden, "Authentication and its privacy effects", IEEE Internet Computing, Nov. 2003.

[6] D. Nayak, N. Rajendran, D.B. Phatak, V.P Gulati, "Security issues in mobile data networks", Vehicular Technology Conference VTC2004-Fall, 26-29 Sept. 2004.

[7] Heikki Kaaranen, Ari Ahtiainen, Lauri Laitinen, Siamak Naghian, and Valtteri Niemi, UMTS Networks: Architecture, Mobility and Services, 2nd Edition, John Wiley & Sons, Ltd., 2005.

[8] Valtteri Niemi, Kaisa Nyberg, UMTS Security, John Wiley & Sons, Ltd., 2003.

[9] Wenbo Mao, Modern Cryptography: Theory and Practice, Prentice Hall PTR, 2003.

[10] Matt Bishop, Computer Security: Art and Science, Addison Wesley Professional, 2002.

[11] Peter Howard, "3G security overview", In Proceedings of the IIR Fraud and Security Conference, March, 2000.

[12] Geir M. Koien: An Envolved UMTS Network Domain Security Architecture, 2002.09.05.

[13] OMNeT++: http://www.omnetpp.org/

Biography

Győző Gódor received the MSc degree in electrical engineering in 2003 from Budapest University of Technology and Economics (BUTE), Budapest, Hungary. He is currently a PhD student in the School of Electrical Engineering at BUTE in the field of Telecommunications. His research interests include mobile and wireless network security and reliability, interworking between various networks and protocol failures analysis. He is a student member of the IEEE, IEEE Computer Society and the IEEE Communications Society.

Sándor Imre graduated from the Technical University of Budapest (TUB) in 1993. He received his dr. univ degree in 1996 and the Ph.D. degree in 1999. He is associate professor at Department of Telecommunications of TUB. His main interests include different fields of mobile and wireless communications such as CDMA systems, software defined radio, mobile security etc.