GUAP - A STRONG USER AUTHENTICATION PROTOCOL FOR GSM

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE OF BILKENT UNIVERSITY

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

By

Özer AYDEMİR

January, 2005

Bilkent University, ANKARA

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science

_____
Assist. Prof. Dr. Ali Aydın SELÇUK (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science

_____
Assist. Prof. Dr. İbrahim KÖRPEOĞLU

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science

_____
Prof. Dr. Varol AKMAN

Approved for the Institute of Engineering and Science:

_____
Prof. Dr. Mehmet B. BARAY
Director of the Institute Engineering and Science

ABSTRACT

GUAP – A STRONG USER AUTHENTICATION PROTOCOL FOR GSM

Özer Aydemir

M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. Ali Aydın SELÇUK
January, 2005

Traditionally, the authentication protocols for cellular phone networks have been designed for device authentication rather than user authentication, which brings limitations and restrictions on the functionality of the system. In this thesis we propose a user authentication protocol for GSM (Global System for Mobile) based cellular phone networks. Our protocol permits the use of weak secrets (e.g. passwords or PINs) for authentication and provides certain flexibilities for GSM users. The simulation results on currently established user authentication protocols and GUAP are presented. Our proposal also has a capture resilience extension to disable captured cellular phones securely.

Keywords: wireless network security, user authentication, GSM, strong password protocols, capture resilient

ÖZET

GUAP – GSM İÇİN KUVVETLİ KULLANICI ASILLAMA PROTOKOLÜ

Özer Aydemir

Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. Ali Aydın SELÇUK
Bilgisayar Mühendisliği Bölümü, Bilkent Universitesi, Ankara
Ocak, 2005

Hücresel telefon ağları için asıllama sistemleri geleneksel olarak kullanıcı asıllaması yerine sistemin kullanımı üzerine kısıtlamalar ve sınırlar getiren cihaz asıllamalarını kullanmak üzere tasarlanmışlardır. Bu tezde biz kullanıcı asıllamasını GSM'e uygulayan bir protokol tasarladık. Bizim protokolümüz zayıf kullanıcı şifrelerinin kullanılmasına müsaade etmekte ve GSM kullanıcılarına çeşitli esneklikler sağlamaktadır. GUAP ve literatürdeki ana kullanıcı asıllaması protokollerinin similasyonları da aynı zamanda bu tezde yer almaktadır. Protokol aynı zamanda ufak bir değişiklikle çalınmalara karşı hesapların güvenliğini sağlayabilmektedir.

Anahtar Kelimeler: Kablosuz ağ güvenliği, kullanıcı asıllaması, GSM, kuvvetli şifre protokolleri çalınmaya karşı dayanıklılık

İstanbul'a

(AYRI GEÇEN YEDİ YIL İÇİN)

Acknowledgement

I would like to express my gratitude to my supervisor Assist. Prof. Dr. Ali Aydın SELÇUK for being a light on the hazy way of my graduate study. His trust and encouraging comments in the supervision of this thesis was the key point for reaching the end. I am honoured to be his first graduate student.

I would like to express my special thanks to Assist. Prof. Dr. İbrahim KÖRPEOĞLU for lecturing us Wireless Communications and showing keen interest to the subject matter and accepting to read and review the thesis. Kindfull thanks to Prof. Dr. Varol AKMAN for his kind and valuable advices about graduate study and life during both my graduate and undergraduate life, for being a perfect instance in DAD101 class for candidates and also for accepting to read and review the thesis.

I would like to thank Ata Türk and Kamer Kaya for their kind helps during my graduate study, thanks to my homemates Oğuz Kurt and Ali Buğdaycı for their morale support.

I thank TÜBİTAK UEKAE İLTAREN Research Group for highly motivating me during my graduate study.

Finally I thank to my mom, dad and brother for staying patient at İstanbul for six years during my undergraduate and graduate study in Ankara.

CONTENTS

1. INTRODUCTION 1

1.1 PROBLEM DESCRIPTION 2

2. RELATED WORK 5

2.1 EVALUATION OF AUTHENTICATION 5
2.2 STRONG PASSWORD PROTOCOLS 6
2.2.1 Encrypted Key Exchange 8
2.2.1.1 RSA-EKE 9
2.2.2 Augmented Encrypted Key Exchange 10
2.2.3 Protocol of Gong et al. 11
2.2.4. SPEKE 12
2.2.5 SRP 13
2.2.5.1 The role of u in SRP 14
2.2.6 OKE 15
2.2.7 Zhu et al.'s protocol 15
2.2.8 PDM 16
2.2.9 Others 17
Protocol for Capture Resilient Devices 17
Kaliski and Warwick's protocol 20

3. SECURITY ANALYSIS 21

3.1 GONG ET AL.'S PROTOCOL 21
3.2 E-RESIDUE PROBLEM OF RSA 23
3.3 E-RESIDUE AND OKE 25
3.3.1 E-residue Attack on OKE 26
3.4 SOLUTIONS TO E-RESIDUE PROBLEM 27

4. WIRELESS SECURITY AND GSM 29

4.1 SECURITY ISSUES ON WIRELESS SYSTEMS 29
4.1.1 Security Issues on Mobile Phones 29
4.1.2 Security in Handheld Devices 31
4.2 GSM 32
4.2.1 PIN Code Protection 33
4.2.2 GSM Authentication 34
4.2.1.1 GSM Encryption 36
4.2.1.2 Cryptographic Algorithms of GSM 36

5. GSM USER AUTHENTICATION PROTOCOL (GUAP) 37

5.1 EXTENSION FOR KEY DISABLING 40

6. SIMULATIONS 42

6.1 ENVIRONMENT 42
6.2 SIMULATION RESULTS 43
6.2.1 Results on HLR 44
6.2.2 Results on VLR 46
6.2.3 Results on Mobile Client 47
7. CONCLUSION 50
REFERENCES 51

LIST OF FIGURES

FIGURE 1 CHALLENGE RESPONSE PROTOCOL 6
FIGURE 2 ENCRYPTED KEY EXCHANGE PROTOCOL WITH PUBLIC KEY ENCRYPTION... 8
FIGURE 3 GONG ET AL.'S PROTOCOL 11
FIGURE 4 AUGMENTED EKE PROTOCOL 10
FIGURE 5 SPEKE PROTOCOL 13
FIGURE 6 KEY RETRIEVAL PROTOCL 18
FIGURE 7 SIGNATURE PROTOCOL WITH KEY DISABLING 19
FIGURE 8 OPEN KEY EXCHANGE PROTOCOL 25
FIGURE 9 GENERATION OF SSD KEYS IN IS-41 30
FIGURE 10 GSM AUTHENTICATION PROTOCOL 34
FIGURE 11 CONSECUTIVE CONNECTIONS TO GSM 35
FIGURE 12 GUAP PROTOCOL 38
FIGURE 13 INITIALIZATION PHASE FOR KEY DISABLING 40
FIGURE 14 KTOOLBAR SIMULATOR 43
FIGURE 15 AVERAGE SIMULATION RESULTS ON HLR 45
FIGURE 16 AVERAGE SIMULATION RESULTS ON VLR 47
FIGURE 17 AVERAGE SIMULATION RESULTS ON MOBILE CLIENT 49

LIST OF TABLES

TABLE 1 E-RESIDUE RESULTS 23
TABLE 2 SIMULATION RESULTS ON HLR 44
TABLE 3 SIMULATION RESULTS ON VLR 46
TABLE 4 SIMULATION RESULTS ON MOBILE CLIENT 48

1. Introduction

When the door is invented, inventor shouted “Who’s that!” and the authentication has already taken place in human beings life. Messengers brought passwords with their messages to prove that they are a trusted entity of an ally, while an enemy was trying to eavesdrop the passwords or impersonating the messenger. Since that time we are living with three people: Alice, sender of the message, Bob, the receiver, Trudy –the impersonator or eavesdropper.

Since computer systems have became a part of our lives, authentication became a mandatory issue for all systems, and since authentication has become the part of computer systems, trudies have been trying to break the authentication. The weaknesses of authentication systems usually depend on the weaknesses of password holders, where they are able to hold weak passwords. The engineers, scientists and designers have two ways to follow. To make the authentication systems robust even if weak passwords are used, or to make the passwords robust even if weak authentication systems are used. The first solution brings the strong authentication password protocols, and the second solution brings the device authentication systems, where devices replace user entities. Both schemas have advantages and disadvantages; first solution comes with extra computational costs where second solution takes out the flexibility of the systems.

Wireless communication systems have special importance in computer systems. They all are new technologies and may have undiscovered weaknesses. They are transmitting data over the air which is easy to eavesdrop and with cellular mobile phones they all are spread all over the world. Also usually wireless mobile systems have limited capability of computation with their limited computation power and energy.

Mobile telephone networks are becoming more popular everyday, and the *Global System for Mobile* (GSM) is the most commonly used standard for mobile communications with more than one billion users worldwide [6]. GSM defines the services, functional/subsystem interfaces, and protocol architecture for digital mobile radio networks. Identity of a GSM subscriber is established by the *subscriber identity module* (SIM). For authentication, GSM relies on a symmetric encryption key embedded in the SIM card [3, 8, 19].

Those restrictions force the wireless system designers to use device authentication instead of user authentication. In the scope of thesis the strong user authentication protocols will be analyzed and a new strong user authentication protocol for GSM will be introduced with the experimental results on mobile phones, also some extensions like end-to-end encryption for cellular phones and a precaution for loss of cellular phones will be introduced.

1.1 Problem Description

One of the reasons for preferring device authentication in GSM rather than user authentication is humans' inability to remember strong secrets. Human users

tend to choose weak secrets such as short pins, dictionary words, and birthdays as passwords, which are vulnerable to dictionary attacks. However using device authentication in a cellular mobile network brings some restrictions to the users. All users are defined by a SIM card, which holds a strong cryptographic key for the user.

The idea of including user authentication in GSM system is the main motivation of this thesis. Cellular phones are easy to eavesdrop and if user authentication is used in those phones, the authentication scheme has to be resistant to dictionary attacks. The main solution to this problem is strong password protocols, but they are computationally demanding on both client and server sides. In this work we studied modifying the strong password protocols in order to reduce the computational cost on client side while staying resistant against attacks.

We present a new strong authentication protocol for GSM that allows using weak user secrets rather than embedded strong keys. Allowing user dependent keys breaks the dependency to SIM card and brings some flexibility such as redirection of calls, reaching accounts or disabling stolen cards without interacting with the operator of the service provider without the need of SIM card. Another problem for mobile phone users is the loss or stealth of the mobile phone, we make our proposal capture resilient that provides revocation of the lost phone. Also in this thesis some useful extensions to mobile authentication systems will be proposed as easy capture resilience. The

analysis and proposals are supported with the simulation results on mobile phones.

The notations common to the rest of the paper are as follows:

Π_i : Password of user i,

$E_x\{p\}$: Public key encryption of plaintext p with the key of x,

$K\{p\}$: Symmetric key encryption of plaintext p with key K.

2. Related Work

2.1 Evaluation of Authentication

Before explaining the proposed strong password protocols, we would like to show the evolution of the password protocols. The first approach is so naïve that receiver asks, "Who's that?" and Alice gives the plain password. In that case eavesdropper easily gets the password and authentication is only one way,

Alice → Bob: Password of Alice

Next approach is to hash the password in order to hide it from eavesdropper. Alice gives hashed password instead of plain text and Bob applies the same hash function to the password; if two values are equal then authentication is done. Still authentication is one way and now eavesdropper can use the hashed password instead of password also if he wants to get the plain password, checks all dictionary words in a short time to find the corresponding password, which gives the same hash output.

Alice → Bob: H(Password of Alice)

In order not to send only hashed password through channel a challenge-response method is used. Alice sends a random number called 'challenge' to Bob, Bob encrypts the challenge of Alice with password (or hashed password)

and send it with his challenge. Alice decrypts her challenge sent by Bob and checks whether encryption is correct or not if it is correct then sends Bob's challenge back encrypted with password. In that case mutual authentication is achieved, but the scheme is still vulnerable against dictionary attack. Trudy sends a challenge as Alice to Bob. Bob encrypts it and sends back to him. Then Trudy closes the connection and starts to decrypt the message of Bob with dictionary words, when he finds the correct challenge then the password guess is correct.

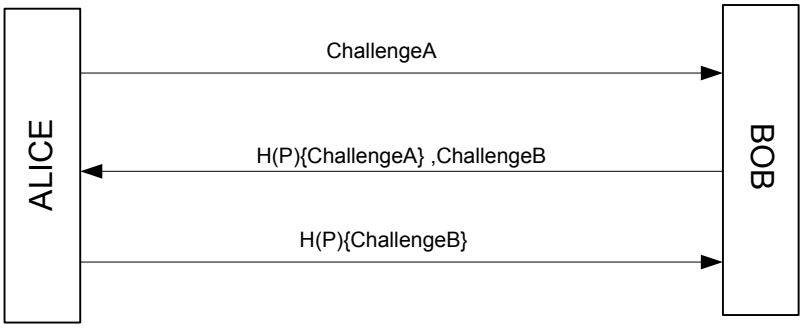

Figure 1. Challenge Response Protocol

2.2 Strong Password Protocols

Since the authentication protocols evolve against dictionary attacks, there are two options to make them resistant: to make the password complex in order not to be in a dictionary, or to make the protocol not leak any information to the Trudy. The first option comes with the basic solution called device authentication, second option needs more complex approach called strong password protocols and since 1992, network security specialists are working on it.

Assume that two parties Alice and Bob try to establish a secret, authenticated session key for their communication. The only secret they share is a user password Π, which is vulnerable to dictionary attacks. The aim of strong password protocols is to authenticate the user while protecting the password against dictionary attacks by online eavesdroppers. Two early works in this category are the Encrypted Key Exchange (EKE) protocol of Bellovin and Meritt [1] and the protocol of Gong et al. [5]. Both protocols aim to authenticate the parties of communication and protect the user's password against eavesdroppers.

EKE is protecting the weak secret by encrypting it by using a public key cryptosystem. In 1993 they published the Augmented EKE (or A-EKE) protocol, which protects the weak password against server database disclosure, in the same year Gong et al. [5] presented a new approach to the protecting poorly chosen secrets from guessing attacks, which also depends on the public key cryptography. Jablon [9] presented simple password exponential key exchange (SPEKE), which uses Diffie-Hellman method for strengthen the weak secret. In 1997 Lucks [14] presented the open key exchange (OKE) that prevents the client to regenerate public, private key pairs per session. In 1998 T.Wu [23] presents secure remote password (SRP), which uses asymmetric key exchange (AKE). In 2000 MacKenzie found a flaw in OKE and brought some constraints on choosing public key. In 2001 Perlman and Kaufman [18] present the password-derived moduli (PDM), which decrease the computations on the server side. In 2002 Zhu et al [24] presented an authentication method for

restricted devices, the solution is based on OKE except, they changed the protection part.

2.2.1 Encrypted Key Exchange

The Encrypted Key Exchange (EKE) [1] protocol provides secure authentication between user and a server using a weak secret. The protocol based on the combination of symmetric and asymmetric cryptography. The main idea is to hiding the weak secrets by hardening it with a public key cryptosystem. The protocol is a generic protocol, which suits for all known public key cryptosystems including RSA, Diffie-Hellman, El Gamal. The protocol generates a strong session key at the end of the protocol and rest of the communication uses this session key for securing the system. There are two main classes of the EKE protocol; one based on public key encryption, the other on Diffie-Hellman key exchange.

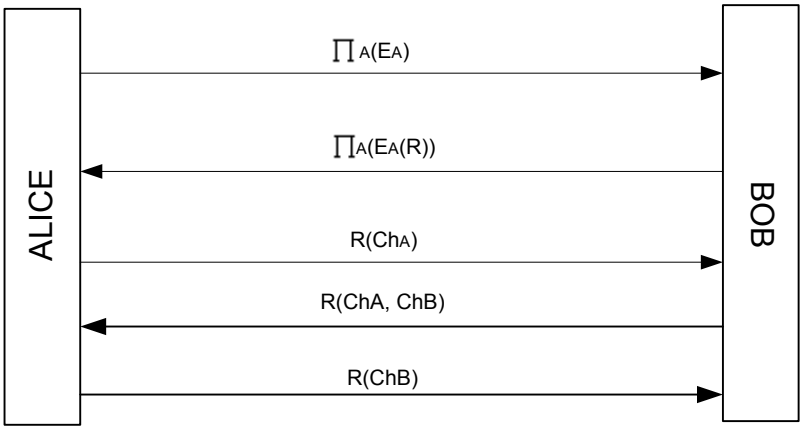

Figure 2. Encrypted Key Exchange Protocol with public key encryption

The protocol in Figure 2 illustrates the EKE protocol with public key encryption. Here Π_A is the password of Alice, E_A is the public key generated by Alice per session, R is a symmetric session key generated by Bob. Alice, who wants to authenticate herself to Bob, generates a public-private key pair for the current session and encrypts the public key with the password and sends it to Bob. Bob, knowing the password Π_A, decrypts the message and obtains the public key. He then generates a random secret R, encrypts it with the public key E_A and Alice's password, and sends it to Alice. Alice decrypts the message to get session key R, and they carry out a challenge-response protocol to authenticate each other. EKE is a generic protocol and can be used with any public key scheme with minor modifications. Even though EKE is a secure user authentication protocol with weak secrets, generating per session public-private key pairs and doing private key operations on client side make it infeasible to use with computationally restricted devices.

2.2.1.1 RSA-EKE

An RSA public key pair is (e, n), where e is the exponent for encryption and n is the modulus, which is product of two large primes. In RSA-EKE, n shouldn't encrypted with password, because eavesdropper may decrypt $\Pi\{e,n\}$ with candidate password Π' and check whether n is a valid RSA public key modulus (if it is prime or has more than two factors, it is absolutely invalid). If n is invalid Π' is incorrect (eliminated from the candidate list). Also e is always odd, if decryption with candidate passwords results on an even e then the candidate passwords is eliminated, so encryption function should add one in

the probability of %50, in order not to leak any information about weak password Π.

El Gamal and Diffie Hellman methods are also proposed with EKE [1] different than RSA-EKE.

2.2.2 Augmented Encrypted Key Exchange

EKE is secure against dictionary attacks but because of making operations with plain password, it is not resistant against password database disclosure. Belovin and Merit [2] proposed a new solution against this problem called Augmented-EKE or A-EKE. A-EKE uses hashed password instead of plain one in protocol and keeps the hashed value of passwords in server database.

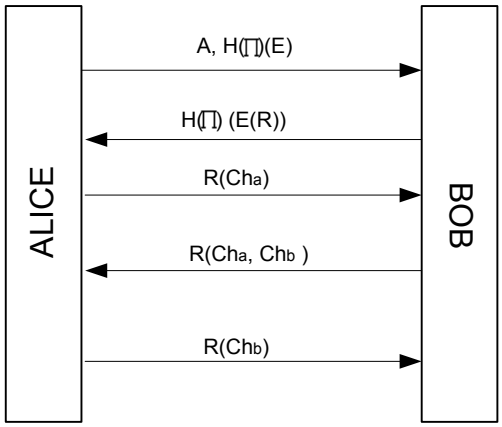

Figure 3. Augmented EKE Protocol

Bob stores the H(Π) as the verifier of the password where g is a generator in the Z_p^* . By this modification A-EKE is a secure authentication protocol for

weak secrets against eavesdroppers, impersonators and database disclosure doesn't result in plaintext passwords anymore.

2.2.3 Protocol of Gong et al.

After EKE was proposed, Gong et al. [5] proposed a different solution to strong authentication with user passwords. The solution contains a third party, which is a trusted center as in Kerberos. The parties in the system authenticate each other by the help of the trusted server. In the protocol below, unlike EKE, there is no need for generating public/private key pair per session, but there is a need for a trusted server whose public key is known by all parties.

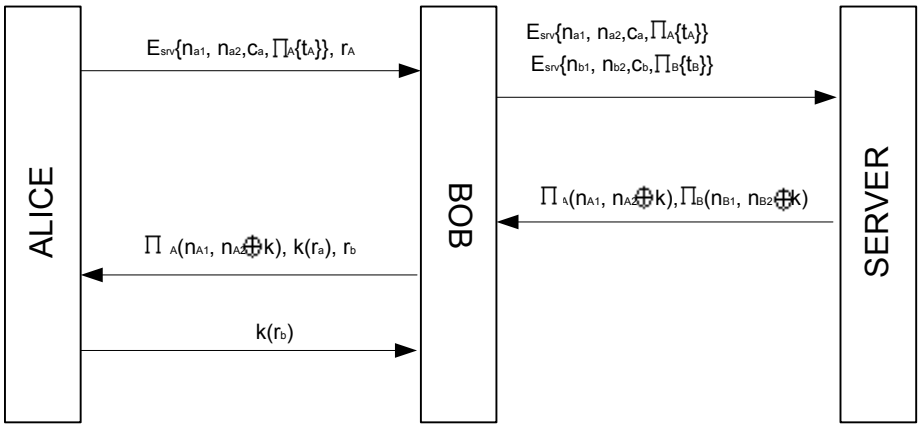

Figure 4. Gong et al.'s protocol

In the protocol above Alice wants to communicate to Bob through an authenticated channel. Alice generates three random numbers n_{A1}, n_{A2}, and c_A, and encrypts the timestamp with her password Π_A then she encrypts all with

the public key of the server and sends them to Bob with a random challenge r_A. Bob generates the same message with his random numbers n_{B1}, n_{B2}, c_B and timestamp t_B, and forwards both messages to the server. The server decrypts both messages and by decrypting the $\Pi_A\{t_A\}$ and $\Pi_B\{t_B\}$ checks whether timestamps are fresh or not; if they are, server generates a session key for the session between Alice and Bob. Server masks the key of Alice and Bob, encrypts the n_{A1} (n_{b1}) and masked session key with Π_A (Π_B) for them sends both messages to Bob. Bob forwards Alice's portion. After decrypting their messages and getting the session key k, Alice and Bob carry out a challenge response protocol to authenticate each other.

After Belovin and Merit proposed the EKE and Gong et al. proposed their protocol, several new strong password protocols are proposed. We will analyze them in the following subsections.

2.2.4. SPEKE

The SPEKE proposed by Jablon [9], is a variant of Diffie Hellman A-EKE, however instead of generator g, SPEKE uses a function F(s) where s is the weak secret. Also SPEKE does not use any symmetric encryption.

According to the proposal, F function should be chosen carefully, in order not to leak any information about password. Offered candidate functions for F(S) are g^S mod p; $S^{p-1/q}$ mod p where q is an order of a generator in G_p; $g_q{}^S$ mod p where g_q is an element of order q; where p and q are large primes and $q|(p-1)$.

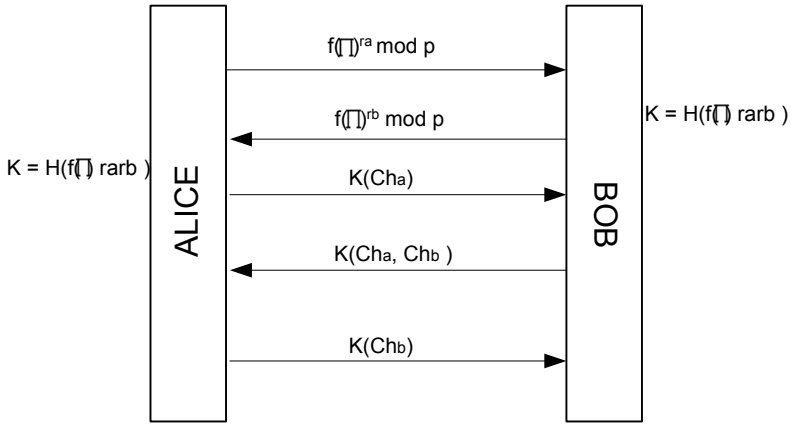

Figure 5 SPEKE Protocol

Π is hidden in the public key so that there is no need to use password as a key in symmetric key operation. Session key is also a function of password and both sides' public key. The same challenge response schema is used as in EKE.

2.2.5 SRP

In 1998, Thomas Wu [23] proposed Secure Remote Password Protocol (SRP), SRP is based on the computation of strong key in two different (and equivalent) ways. There are four functions called S, R, P, and Q. The strength of the protocol is based on the calculation of key K = S(R(P(w), P(x)), Q(y, z)) = S(R(P(y), P(z)), Q(w, x)). A set of candidate functions are given by Wu:

$P(x) = g^x$

$Q(w,x) = w + ux$

$R(w, x) = wx^u$

$S(w,x) = w^x$

SRP Protocol

Server stores a verifier per user and a salt. $v = g^{H(\Pi,s)}$

At the beginning of the protocol server sends the salt to the user

- Alice → "Alice"
- Bob → Sends her salt s
- Alice → computes v and generates 'a'(nonce) and sends g^a
- Bob →generates 'b' (Bob's nonce) Computes $B = v + g^b$ sends B, random u
- Both calculates $S = g^{ab+ubx}$
- Challenge response occurs.

Why $B = v + g^b$

If instead of $B = v + g^b$ $B = g^b$ is used then Imposter of server sends salt to Alice where she previously snooped. Alice sends g^a, Impostor picks b and u Alice computes $S = g^{ab} * g^{ubx}$ sends proof; Impostor alerts the failure and closes the connection. Impostor now has g^a and b and proof of S. He chooses candidate password Π', computes v' and from the construction gets S'. If S' matches with S then candidate password is correct.

2.2.5.1 The role of u in SRP

u is generated in 3rd message. Assume that it is not random and it is always 1 (or known). If an intruder client who captured the v of Alice, impersonate Alice:

- Impostor → Alice
- Bob → s

Impostor computes $g^a v^{-u}$ instead of g^a

- Impostor → $g^{a}v^{-u}$
- Bob = v+ g^{b}
- Impostor computes session key as
  - S = $(B\text{-}v)^{a}$ mod n
  - Because Bob also generates same S because of wrong message sent in the 2nd message.

2.2.6 OKE

In RSA-EKE public key provides the freshness of the sessions, so it is mandatory to generate public-private key pair for each session, which increases the computation of the protocol. Lucks [14] proposed a new solution called Open Key Exchange in 1997. In OKE freshness are guaranteed with a random nonce generated through session and Bob (server) may use the same key pair for each session to each client. It will be suitable for client-server systems, where multiple clients want to authenticate to a server. As will be analyzed in security analysis chapter RSA-EKE and OKE is not resistant against e-residue attack, and author protects the protocol against e-residue attack by adding some extra turns to protocol. In 1999 MacKenzie, Patel and Swaminatham [16] found a new attack on protected–OKE.

2.2.7 Zhu et al.'s protocol

In 2000 Zhu et al.'s [24] protocol proposes a protocol which resembles to protected- OKE but uses a new method to protect the protocol against e-residue

attack. MacKenzie et al's [16] solution to protected-OKE is infeasible for restricted devices, because of heavy public key operations, against this situation Zhu et al. come up with a new protection method called "interactive step"[24]. The step is basically relies on checking the correctness of RSA keys by testing it with random numbers. Receiver of the public key generates m random numbers and encrypts each of them with the public key and sends it to the generator of the keys. Generator decrypts the encrypted numbers and returns them to receiver, unfortunately this solutions comes with the extra public key encryption and decryption operations, which brings the computational cost.

2.2.8 PDM

Network security specialists usually work on decreasing the computational cost on the client side, in order to make the protocols suitable for restricted devices. Kauffman and Perlman [18] focused on the opposite direction and proposed a PDM (Password Derived Moduli) to decrease the computational costs on server side . PDM can be used for both mutual authentication or downloading secret information as private key.

Basically server stores a prime number p, a random number B and the private information of client encrypted with client's password. Prime number p is also generated from password in the user's initialization. Alice chooses a random number A and sends 2^A mod p to Bob. Bob calculates 2^{AB} mod p and 2^B mod p. Sends 2^B mod p and multiplication of secret information with 2^{AB} mod p to

Alice. Alice calculates 2^{AB} mod p and gets her secret information. As authors stated the protocol is heavier than any other strong authentication protocol on client side, however it is so fast for server. So the protocol suits on applications, where server serves to large amount of powerful clients.

2.2.9 Others

Protocol for Capture Resilient Devices

Disabling the key of captured mobile phone is another critical problem for both GSM operators and mobile phone users. MacKenzie and Reiter proposed a new scheme for the mechanism of secure key disabling [15]. The mechanism has two options: without key disabling and with key disabling. The first mechanism provides client to store its private key in the server and retrieve it when is needed. The second mechanism provides clients to sign a message, where both server and client have shares of the private key, second mechanism also support key disabling.

Both mechanisms have a device initialization phase. Figure 6 illustrates the key retrieval protocol. In the initialization phase Alice defines two random numbers ν and a, calculates hashed password as b, defines c as $f(\nu, \Pi) \oplus sk_{dvc}$ where sk_{dvc} is the private key of the client and τ as E_{pksrv} ({a,b,c}). The values ν, a, τ, pk_{dvc} and pk_{server} are saved in the stable storage on the device, other values including sk_{dvc}, Π, b and c are deleted from the device.

During agreement Alice generates a random number ρ, and defines β as hashed password, encrypts both ρ and β with Bob's public key as γ and defines δ as mac of γ and previously initialized τ, sends γ, δ, τ to Bob.

Bob gets a, b and c from τ, checks the mac of γ and τ, gets ρ and β from γ checks whether β equals to b, mask c with ρ as η and sends it back to Alice. Alice unmask η and gets c and gets sk_{dvc} from it.

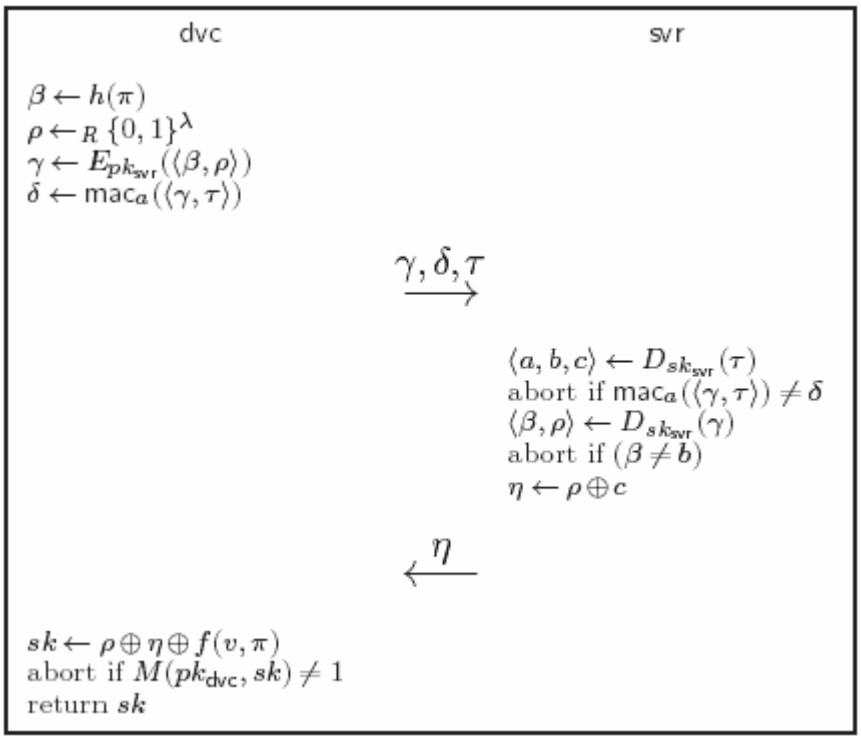

Figure 6. Key Retrieval Protocol

The second scheme is a signature protocol allowing key disabling. In the device initialization phase Alice initialize random numbers t, a, ν, defines u as h(t), b as h(Π) , d_1 as f(ν,Π), d_2 as d-d1 (where d is an RSA private key) and τ as E_{Bob}({a,b,u, d_2, N}) (where N is RSA modulus). The values t, a, ν, τ, pk_{Alice},

and pk_{Bob} are saved on the device, other values including u, b, d, d_1,d_2, Φ(N) are deleted. τ and t are stored in other storable media and registered to the server in order to key disabling purpose.

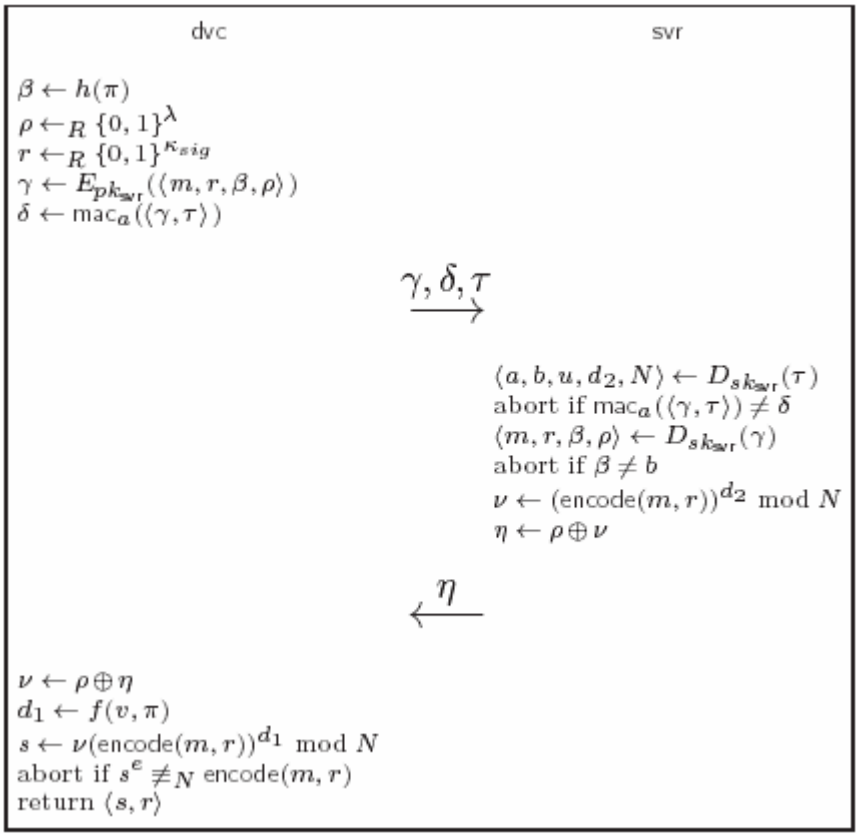

Figure 7. Signature Protocol with Key Disabling

Figure 7 illustrates the protocol. During signature protocol Alice generates two random numbers r and ρ, defines β as h(Π), γ as E_{Bob} (m, r, ρ, β) where m is the message to be signed and defines δ as mac of γ and τ. Alice sends δ, γ and τ to Bob. Bob gets a, b, u d_2, N from τ, checks the mac of γ and τ, gets m, r, ρ, β from γ, checks whether β equals to b, if they equals then calculate m^{d_2} mod N

masks it with ρ and sends ρ to Alice. Alice gets d_1 from password and ν, unmask ρ and calculates $m^{d1}m^{d2} \mod N = m^d \mod N$. The message has been signed with Alice's key where both party doesn't have the exact d instead have its shares.

Alice sends τ and t to the Bob, if the device is captured or lost. Bob gets u from τ and checks whether h(t) is equal to u. If it is equal Bob disables the key of Alice.

Kaliski and Warwick's protocol

Kaliski and Warwick proposed a new scheme for server-assisted strong key generation from user password [4]. Their solution is based on calculating the strong key with the help of a set of terminal clients. The proposed scheme assumes that the clients are mobile, but aren't restricted on neither energy nor space.

Both EKE and A-EKE give the flexibility to the designer by permitting to choose the public key cryptosystem. Diffie-Hellman, RSA and ElGamal are the three offered choices for public key operation in the protocol. SPEKE is designed for Diffie-Hellman, OKE uses the RSA for public key operations, SRP and PDM use discrete exponentiation.

There are still open problems in strong authentication with weak passwords. All the solutions depend on asymmetric cryptography and this brings the computational heaviness.

The main aim to that problem is to use RSA and use small e for decreasing the computation in one side because of encryption with small e is much lighter than the decryption in RSA. However e-residue attack is a main obstacle against this solution and precautions against e-residue attack is a considerable issue for restricted devices. For those reasons, proposed strong authentication schemes are not suitable for restricted wireless clients.

3. SECURITY ANALYSIS

In the core of this thesis, we focus on the EKE, Gong et al's protocol, OKE and Zhu et al's protocol. This chapter analyzes the security of those protocols.

3.1 Gong et al.'s Protocol

In the first round of the protocol Alice generates n_{a1}, n_{a2} and $c_{a,}$ and Bob generates n_{b1}, n_{b2} and $c_{b.}$ They both put those random numbers in their first message to trusted server.

Server passes the session key to Alice and Bob in the message of $\Pi_A(n_{a1}, n_{a2} \oplus k)$ and $\Pi_B(n_{b1}, n_{b2} \oplus k)$. If Alice or Bob connects to server through multiple connections n_{a1}, and n_{b1} provides the information of which session key k

belongs to which session. Also it brings the freshness guarantee for Alice and Bob.

Usage of n_{b1} protects Alice's password against Bob and both password against eavesdropper. If server passes key without blinding with n_{a2} (or n_{b2}) Bob may get the k from his portion and choose a candidate password Π_A' from a dictionary and decrypts the Alice's portion of the message, if he gets the same key k from Alice's password then the password guessing is valid. Also eavesdropper may generate a dictionary of candidate password couples for Alice and Bob (with the size of N^2 where N is the size of standard dictionary), with those candidate passwords eavesdropper decrypts the both $\Pi_A(n_{a1},k)$ and $\Pi_B(n_{b1},k)$, if both decryption gives the same output then eavesdropper is successful on guessing passwords.

Assume that c_a and c_b are not used on the protocol. If a previously used session key compromised and all message of the session is saved by eavesdropper then eavesdropper may chooses a candidate password Π_A' and gets n_{a1}' and n_{a2}' with a guessed timestamp t' Trudy may construct $E(n_{a1}',n_{a2}',\Pi_A'(t'))$ if a constructed message is equal to first message of the protocol then password guessing is correct. For this reason c_a is used to protects to regeneration of first message by the help of the compromised key.

3.2 E-Residue Problem of RSA

Public key cryptosystem for RSA has a public and private key pair of large natural numbers. Public key is (e, n): n is the product of two large primes p and q. Public exponent e is relative prime to $\phi(n)$ (where $\phi(n) = (p-1)*(q-1)$). Private decryption key (d) is calculated by $ed = 1 \bmod \phi(n)$.

A message m is encrypted by raising m to power of e on modulus n ($c = m^e \bmod n$) and decrypted raising cipher to power of d on modulus n ($m = c^d \bmod n$). In RSA there are two domains of numbers: plaintext domain and ciphertext domain and the number of elements in both domains are equal to n. Encryption is a one-to-one and onto function from plaintext domain to ciphertext domain and decryption is a one-to-one and onto function from ciphertext domain to plaintext domain.

The correctness of RSA is depends on the correctness of $e*d = 1 \bmod \phi(n)$. The same functions with the property of $e*d = z \bmod \phi(n)$ (where $z > 1$) breaks down the one-to-one and onto property of RSA. If e*d is equal to z instead of one, then encryption function covers only a subset of ciphertext domain, and if n is a combination of more than two primes that subset becomes smaller.

E*d mod n	p	q	d	e	subset size	space size	ratio
106	101	107	7102	3	101	10807	0.01
100	101	107	7100	3	107	10807	0.01
50	101	107	3550	3	161	10807	0.01
53	101	107	3551	3	302	10807	0.03
1	101	107	7067	3	10807	10807	1.00

Table 1. E-Residue results

If a message m is encrypted with a fake RSA key, then the encryption result always falls on the proper subset of ciphertext domain. If Alice encrypts the result of encryption with her password then Trudy choose a candidate password Π' and decrypts the message, if encryption covers 1/t of the domain and candidate result falls in the same domain then password guess is 1-1/t percent correct.

Table 1 shows an example of e-residue results on RSA. Results are taken with $p = 101$, $q = 107$, $e = 3$ and $n = 10807$. On each try corresponding d is recalculated. If e*d equals one then the subset space covers all of the space as expected. If it is equal to 106 subset space covers only %1 of total space, which means password guess is %99 percent correct. If the primes p and q increase into large numbers than more effective results can be found.

First two message of the EKE are $\Pi(E_A)$ and $\Pi(E_A(R))$, authors state that one of the encryption with password Π is redundant and only one encryption with password is sufficient. However RSA-EKE has a special property that second message must not be encrypted with password against E-residue attack.

On the point of view of restricted devices, regeneration of per session RSA key pair is a costly operation for both restricted devices and server and also in cellular phone systems home location register (HLR) is a common over a wide area, whose public key may be known by all entities. For this reason

encryption of permanent public key with password is subject to dictionary attack and encryption of $E_A(R)$ is subject to e-residue attack.

3.3 E-Residue and OKE

Open key exchange is the first variant of EKE that eliminates the need of encrypting public key as desired in restricted devices. Figure 8 illustrates the Open Key Exchange Protocol, Alice and Bob agrees on a common secret Π. Alice generates an RSA key pair and a message m and sends m and public key (e,n) to Bob. Bob creates a random numbers a and μ. Calculates p as a hash of e, n, m, μ, Π and q = E(a)*p sends μ, q to Alice. Alice calculates p in the same way, and gets a′ from q computes r = $h_1(a')$ and sends r as a response of getting correct a. Bob checks the response of Alice, calculates session key as k = $h_2(a)$ and his response as t = $h_3(a)$ where h_1, h_2 and h_3 are different hash functions. Sends the response t to the Alice. Alice checks the response and if it is correct then calculates session key and Alice and Bob start to communicate in secure session with key k.

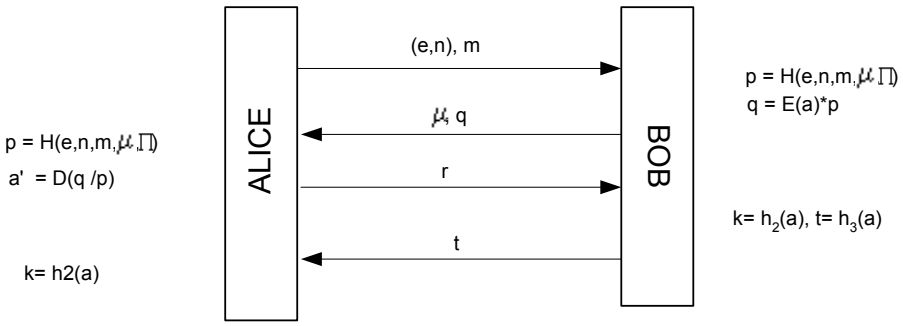

Figure 8. Open Key Exchange Protocol

3.3.1 E-residue Attack on OKE

An impostor may impersonate Alice and send a fake {e,n} pair where e.d is not equal to one. Bob calculates q and sends q and μ to impostor. Impostor closes the channel. And choose a candidate password Π′, computes candidate p′ and gets E(a). If E(a) is a member of subset domain then password guess is probably correct (with high probability) else password guess is wrong.

Solution to that the author proposal is adding a protection calculation on μ before the first message of Bob. Instead of μ Bob generates μ_{-1} and μ_0, and computes $\mu_i = E(\mu_{i-2}.H(\mu_{i-1}))$ and sent back μ_{k-1} and μ_k. Uses both μ_{-1} and μ_0 in the calculations of p. Alice has to make k correct RSA decryption operation in order to get correct values of μ_{-1} and μ_0.

Lucks solution remained valid until MacKenzie, Patel, Swaminatham [16] broke the protection calculations in 1999. For instance if attacker chooses e as 3, n is a large prime and e is relatively prime to n-1. Bob calculates p and sends out q, μ_{k-1} and μ_k. In order to cover μ_s until μ_2 and μ_1. Attacker will decrypt μ_i by solving three cubic roots of μ_i. Then will multiply each root with $H(\mu_{i-1})^{-1}$ to get the three possible solution for μ_{i-2}. Only one of the solutions will be cubic residue over n. Intruder can easily identify the cubic residue by $\mu_{i-2}^{\,n-1/3} \equiv 1$. μ_{-1} and μ_0 are random so it is impossible to eliminate candidates, however they are only nine pairs of (μ_{-1}, μ_0). If for all pairs all possible solutions for E(a) is not a cubic residue then password guess is absolutely wrong.

MacKenzie et al. [16] proposed a new mechanism instead of Luck's solution, however their system forces e to be a large number, which is not suitable for restricted devices.

Zhu et al. [24] proposed a new variant of OKE in 2000. Their protocol is equivalent to standard OKE but the protocol has a new round called "interactive step". In interactive step Bob sends m numbers each encrypted with e and asks to Alice to retrieve the numbers. Authors advice that m should be about twenty, however in that case Bob makes m extra RSA encryption and more importantly Alice makes m extra RSA decryption, which is an undesired situation for restricted systems.

3.4 Solutions To E-Residue Problem

There are two alternatives of the system that we found. The first one is on calculation of q. Instead of calculating q as E(a)*p, Bob may calculate q as E(a*p). In that case if e,n are fake RSA keys than q will always be in the e-residue subset and leak no information about password. However in that case if Trudy impersonates Bob, he will attack on password. Trudy sends a random number x instead of q. Alice calculates a fake a′ and responds to Trudy. Trudy closes the connection and creates a candidate p′ from password, and from the response tries to extract p from x. For this reason this solution may be used if and only if Bob cannot be impersonated.

The second solution is about to harden the e-residue attack. Instead of calculating q as E(a)◊p, Bob calculates q as E(E(a)*p). In that case q will be always in e-residue subset, if key pairs are fake. If e-residue subset is $1/t^{th}$ of the ciphertext domain, then there would be on average t candidate E(a)*p. In that case for all t candidate decryption intruder has to check whether E(a) is in the e-residue subset or not. If all E(a)s are out of the set password guess is incorrect otherwise, it may be correct or not. In that case elimination of candidate password is much more hard.

Both RSA-EKE and OKE are vulnerable against e-residue attack. EKE prohibits the Π(E(R)) operation on RSA version. OKE provides a solution to the e-residue attack, which unfortunately needs several RSA operations on both client and server side and not secure as broken by McKenzie et al. Also Zhu et al. presents a solution to e-residue attack, which depends on the testing the correctness of generated public key pair. It is also inconvenient for restricted devices.

4. Wireless Security and GSM

4.1 SECURITY ISSUES ON WIRELESS SYSTEMS

Wireless systems contain all vulnerabilities of wired systems, plus they may have extra vulnerabilities according to their physical behavior. It is so easy to follow the information traffic without being spotted by the system owners. Everybody may capture the radio signals with the suitable equipments over air. Because of the wireless devices are usually mobile, they have less storage capability and memory. Also network bandwidth of the wireless systems is relatively smaller than wired systems. As a result, codes working on mobile devices should be located in a small portion of the hard drive and use less memory, also because of the less computation power cryptographic operations should be selected carefully. For the reason we stated above, elliptic curve cryptography suits on restricted mobile devices. Elliptic curve cryptosystem needs less storage and computation for equivalent security level than other alternatives like RSA.

4.1.1 Security Issues on Mobile Phones

In second generation systems (2G) authentication is much more important than privacy, because authentication is a mandatory issue in the standards; however, people usually don't want to pay more for privacy.

First generation cellular systems are using Analog Mobile Phone Standard (AMPS). In the 2G systems this standard has some modifications on it and called New AMPS (or NAMPS). GSM is widely used especially in Europe and Australia, US commonly uses TDMA and CDMA (IS-41) (Time and Code Division Multiple Access). All the standards stated above use secret key operations in cryptographic functions. Two main reason of the non-usage of public key operations are:

- The processors of mobile devices are small and restricted.
- The data transfer rate of the mobile devices is small

KEYS

GSM and IS-41[1] use secret key encryption and decryption in cryptographic operations. IS-41 keys are 64 bits and GSM keys are 128 bits. GSM derives session keys and authentication signatures from key, on the other hand IS-41 generates two different shared secret data from key (SSD-A, SSD-B). SSD-A is used as authentication signature and SSD-B is used for session key.

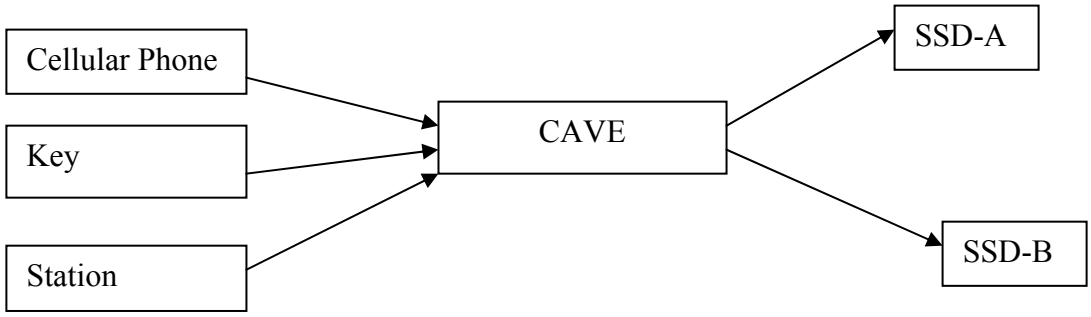

Figure 9. Generation of SSD keys in IS-41

[1] IS-41 is another cellular phone standard uses CDMA and popular especially in North America

The CAVE function is a hash function that works by using a shift register driven over the input data and a somewhat random table, and shuffling the inputs. It takes 23 octets of input and produces 16 octets of output. The output of the CAVE functions are identity of the cellular phone, key of the client and a random number generated from station [20].

In GSM challenge-response processes are unique to operator and identity of the device (International Mobile Subscriber Identity –IMSI), key of the mobile client and cryptographic operations are embedded into a smart card called SIM card. Next chapter will analyze the authentication system of the GSM in details.

4.1.2 Security in Handheld Devices

The main vulnerability on hand held devices is impersonation. Handheld devices usually use device authentication, however, main precaution against impersonation is to use user authentication. Usually hand held devices authenticate user by personal identification number (PIN) that is typically a four digit number.

Palms and PDAs are usually different then cellular phones, where they can work standalone and don't need to authenticate themselves to a center or access point, however they may be extended in order to be a part of a network and to authenticate themselves to a station. Palm usually uses elliptic curve

cryptography in authentication. The key size is about 163 bits, which is as strong as 1024 bit RSA. [17]

4.2 GSM

In 1982 CEPT (Conférence Européene des Postes et Télécommunications) created the GSM (Groupe Spécial Mobile – Global System for Mobile) committee in order to specify the standard for the European cellular systems. In 1988 GSM became a Technical Committee of European Telecommunications Standards Institute (ETSI). Initially GSM was established on 900 Mhz band, In 1988 at the request of the United Kingdom a version of GSM, operating on 1800 Mhz band, was included in the specification. A GSM protocol has three entities: Mobile Station, Base Station and Center.

Mobile Station

GSM subscribers use the mobile stations to make and receive calls. Mobile stations are the combinations of Subscriber Identity Mobile (SIM) and mobile equipment. SIM card is a smart card that is used to deploy subscriber identity into mobile equipment. Each SIM card contains the 15 digits International Mobile Subscriber Identity (IMSI) number, the corresponding to the secret key of the subscriber, the cryptographic functions used in GSM (A3, A5, A8), and the language information of the mobile equipment. First three digits of the IMSI are the country code of the mobile (MCC). Next two digits are mobile

network code (MNC), remaining ten digits are the identification number of the mobile subscriber (MSIN).

Base Station (Visiting Location Register-VLR)

Mobile Subscribers communicate with a base transceiver station (BTS) over radio interface. Base station generally takes the up-link radio signals from MS and converts it into data for transmission to other machines within the GSM network and vice versa.

GSM Operation Center (Home Location Register – HLR)

The center is responsible for accepting mobile subscribers to the system, route the communications between mobile subscribers. HLR is a main database of subscriber information of the GSM operator. It interacts with mobile switching center, which is a center call and control processing [21].

The cellular phone systems have several security aspects. The first step of the aspects is to authenticate human subscriber to SIM card, then authenticate SIM card to GSM operator, and the last encryption of the communication.

4.2.1 PIN Code Protection

The most basic level of the protection in GSM is to protect the mobile device against fraudulent usage by a Personal Identification Number (PIN) code,

against illegal usage of stolen SIM cards. PIN takes a four to eight decimal digit code. SIM also has a second PIN property called PIN2 to activate certain features to the subscriber. After three incorrect attempts to PIN number, the SIM card will be blocked. There is a PIN unblocking key (PUK) which is also stored in the SIM. After 10 incorrect attempts to PUK, SIM card will be blocked permanently.

4.2.2 GSM Authentication

GSM contains three entities in a session: a mobile subscriber (cellular phone), visiting location register (VLR, base station), and home location register (HLR). Alice's SIM card contains a secret authentication key K_A and unique "International Mobile Subscriber Identity" (IMSI). A3, A5 and A8 algorithms are used in authentication, where A3 and A8 are one-way functions and A5 is a symmetric encryption function [8].

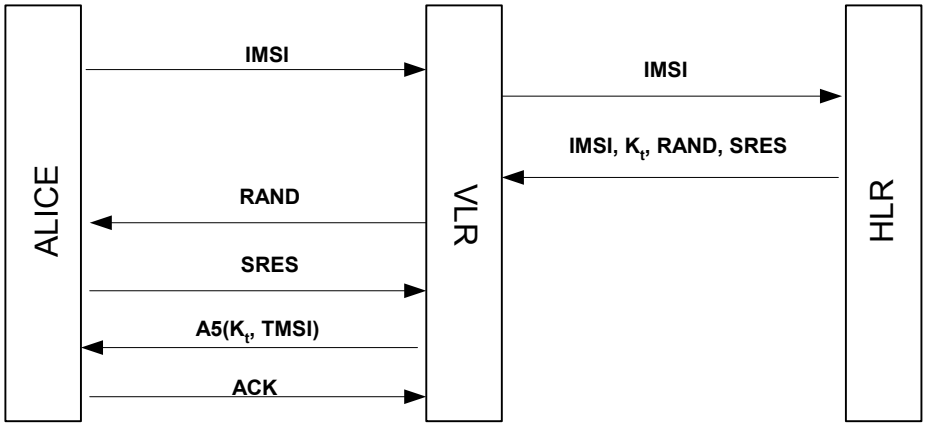

Figure 10. GSM Authentication Protocol

Figure 10 illustrates the authentication protocol of GSM for the first connection attempt of the mobile subscriber to a certain VLR. K_A and K_t respectively are permanent and temporary key of Alice (mobile client). Alice sends her unique identity to VLR, VLR passes this identity to HLR in order to inform it that Alice wants to log in to the system. HLR generates a random number RAND, calculates temporary authentication key K_t for consecutive attempts[2], and the *security result* SRES that is equal to K_A and RAND encrypted with A5. VLR passes RAND to the mobile client and keeps K_t and SRES. Alice calculates SRES and sends it to VLR. If SRES sent by Alice is equal to SRES sent by HLR then VLR sends K_t and TMSI to the mobile client to be used in consecutive authentication attempts without the need of contacting the HLR.

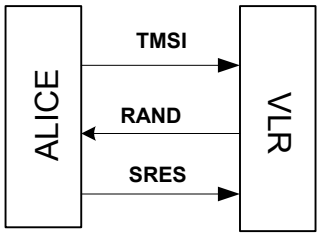

Figure 11 Consecutive Connections to GSM

In Figure 11, consecutive connections of the mobile client to the same VLR is shown. TMSI is the Temporary Mobile Subscriber Identity given to the mobile client. Here, instead of IMSI, mobile client sends TMSI to VLR. VLR generates a random number RAND and sends it to the mobile client. Mobile

[2] K_c is an output of A8 function seeded with K_i.

client calculates the new SRES with temporal key K_T that it received in previous session.

4.2.1.1 GSM Encryption

After authentication of the mobile subscriber, the communications between the system and the subscriber must be protected against fraudulent access. This is provided by encrypting the data on the radio interface using a key K_c and the A5 encryption algorithm. There are up to seven variants of A5 and mobile subscriber and operator agree on one of the A5 algorithms [21].

4.2.1.2 Cryptographic Algorithms of GSM

There are three common encryption algorithms used in GSM security: A3, A5, and A8. A5 is a stream cipher used for encryption in GSM, A3 and A8 are one-way functions take place in the authentication phase.

A3 algorithm is used by a GSM network to authenticate the mobile subscriber. It is a one way function implemented in the SIM [25].

The A5 is the algorithm used for encryption in GSM mobile phones. It can be used on both voice and data connections. It is a stream cipher that uses a 64-bit secret key. A5 is designed to be efficiently implemented in hardware.
There are two versions of the A5 algorithm:
A5/1, which is used in Europe, and A5/2, which is used in export systems
A8 is used to exchange a session key that can be used to encrypt voice or data.
A8 is also one-way function implemented in SIM [25].

5. GSM USER AUTHENTICATION PROTOCOL (GUAP)

In this section, we describe a new user authentication approach for GSM. The current GSM authentication scheme uses the device's authentication key, which is embedded in the SIM card of the user. With the new approach, Alice can use her password instead of the embedded key. Using passwords instead of embedded keys breaks the dependency on the SIM card during authentication. Users will be able to reach their accounts without SIM cards, via any cellular phone, Internet, or a special network. Users can reach their address book, redirect their calls, or get their personal information without the need of either SIM card or giving their personal information to operators of the service provider.

GSM authentication protocol resembles the approach of Gong et al. [5] in certain ways. Both schemes are based on three entities, and in both cases the third entity is a trusted server whose public key is known by all parties. Unlike Gong et al.'s protocol, in GSM authentication, VLR is an automated non-human entity, which is able to remember strong secrets. Another difference is in regard to the use of timestamps. Clock synchronization may be a crucial problem in GSM authentication, which can be solved by generating random nonces for freshness guarantee of the sessions.

Our protocol is illustrated in Figure 12. Mobile user wants to be authenticated to HLR via VLR, using her password Π. A random nonce, RAND, is generated by VLR per session and provides freshness guarantee for the session. Three

random nonces generated by the mobile client are n_1, n_2 and c; n_1 proves the correct decryption of HLR in the fifth message, n_2 masks the session key k, c protects the first message against replay by adversary.

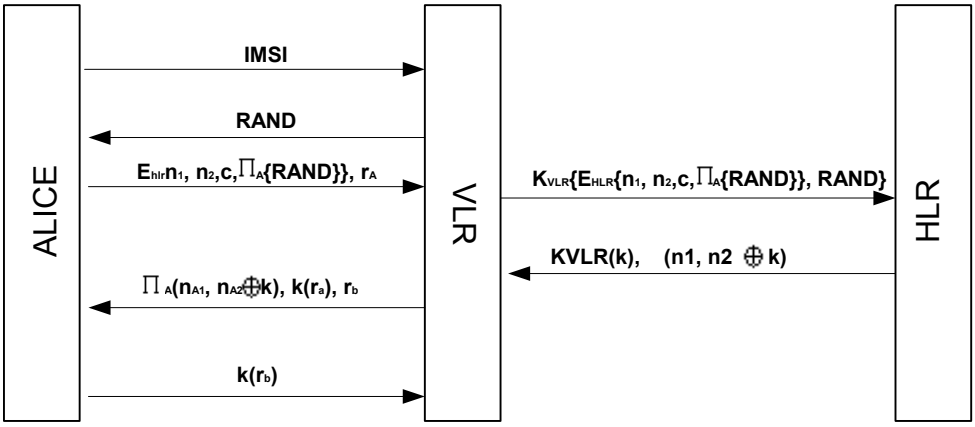

Figure 12 GUAP Protocol

The protocol starts with the client's authentication request by sending its unique identity (IMSI) to VLR. VLR generates and sends a random number RAND to Alice. Alice generates three random nonces n_1, n_2 and c, and encrypts RAND with her password. She then encrypts n_1, n_2, c, and Π(RAND) with HLR's public key and sends it to VLR with a random challenge r_A. VLR takes the message and encrypts the HLR's portion of the message and RAND with its symmetric key, and sends it to HLR. HLR, knowing VLR's symmetric key, decrypts the message, then asymmetrically decrypt the message come from Alice, finally decrypts Π(RAND) to get RAND, if both RAND are equal then HLR is sure about VLR's and Alice's identities. HLR generates a session key for VLR and Alice, encrypts it with VLR's symmetric key for VLR, and

encrypts the masked session key ($n_2 \oplus k$) and n_1 with Alice's password, then sends both messages to VLR. VLR decrypts its portion of the message to get session key k, encrypts the challenge r_A with k sent by Alice in first message, forwards Alice's portion of message with the response to her challenge and a new challenge r_B. Alice decrypts the message coming from HLR and gets the session key k. She then responds to VLR's challenge. In consecutive sessions Alice and VLR can use the generated session key k without need of re-authentication.

In the protocol, the existence of the correct n_1 value in the fifth message indicates that it is the HLR that has decrypted the first message and sending this output. The random nonce n_2 protects HLR's response encrypted by Π against dictionary attacks on Π by an attacker who gets to know k or by VLR. The issue here is a dictionary attack by someone who knows k and hence can guess n_1 and n_2. Random c protects first message against regeneration by VLR: Again a malicious VLR or an adversary that has compromised a past session key k, can choose a candidate password Π' and decrypt the message of mobile client to get candidate $n_1'$and n_2'. Without the confounder c, the adversary can generate a candidate first message. If the candidate message is equal to real message then the password guess is correct [5].

In the protocol VLR is not a user entity so it can remember and perform its operations with its strong secret key K_{vlr}. This reduces the computational cost of Gong et al.'s protocol. The only asymmetric key operation done by the mobile client is a public key encryption in the first message. If RSA is used

here, then the public exponent of the key pair can be fixed to a small prime, reducing the computational cost on the client side.

5.1 Extension for Key Disabling

Loosing cellular phones is a common problem for users. The user has to disable his account against unauthorized usage. Current systems do not provide automatic key disabling after the loss of a cellular device. The user has to call the service provider's operator and prove his identity to disable his SIM card and reveal his private information. In [15] MacKenzie and Reiter proposed a method for capture resilience in networked cryptographic devices where networked devices can sign or encrypt a message through an untrusted server without revealing its private key. Their protocol also provides a key disabling feature for captured devices. The approach assumes an untrusted server and this makes the protocol relatively expensive. In GSM authentication HLR is a trusted identity, hence capture resilience property can be achieved less expensively.

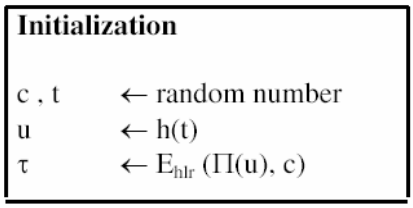

Figure 13 Initialization phase for key disabling

In the initialization of mobile client's account, mobile client generates two random numbers t and c, takes the hash of t as u, and creates a key disabling

ticket τ as E_{hlr} (Π(u), c). Random c and u is deleted and t and τ is taken out of the phone to a storable medium.

After a loss or capture occurs, user immediately sends τ and t to the HLR. HLR decrypts τ with its private key and decrypts Π (u) to get u. After getting u, HLR checks whether u is equal to h(t) or not; if they are equal then the account of the user is disabled for further access. In the ticket the random number c is for protecting the password against dictionary attacks on τ. If c does not exist, an adversary seeing t can generate u and encrypt it with a candidate password and the public key of the server. If it is equal to τ then the password guess is correct.

6. SIMULATIONS

There are two main user authentication protocols for strong user authentication: EKE protocol and Gong et al's protocol. OKE is an efficient variant of EKE where per session key generation is not necessary and client side may compute only lightweight RSA encryption, however it is proven to be vulnerable against e-residue attack. Zhu et al's proposal is a variant EKE that resembles OKE, the only difference is the interactive protocol of Zhu et al's protocol and designed for restricted devices. GUAP is a variant of Gong et al's protocol that is specific to GSM and decreases the computation on VLR and mobile client. In order to show the practical results on the computation time of the protocols we have made a simulation on EKE, Gong et al's protocol, Zhu et al's protocol, OKE and GUAP. Following sections describe the protocol environment, used libraries, results and interpretation of the results.

6.1 Environment

The simulations are implemented in JAVA language. Server side modules are implemented by J2SE v1.4 the mobile client modules are implemented in J2ME 2.0. HLR and VLR codes run on Intel PIV 2.4 Ghz machine with 512 MB RAM, mobile codes run on Sun Microsystems's simulator KToolbar.

Figure 14 KToolbar Simulator: KToolbar is a mobile phone simulator, where developer may run their MIDP (Mobile information device profile) applications.

Cryptography and Network Security API of J2SE hasn't been ported to J2ME yet. For this reason cryptographic functions are used from the Bouncy Castle cryptographic API (www.bouncycastle.org) on both server and client side.

6.2 Simulation Results

All protocols are run for 22 times and computation times are recorded for 512 and 1024 bit RSA keys. JAVA calls the garbage collector in undefined periods and this may take extra times for this reason the worst and the best results are neglected for each simulation and others are presented because both J2ME and

J2SE calls garbage collector of the JAVA and this may effect the computation time of the protocol. Zhu et al's protocol implemented to test ten numbers in interactive step, and OKE implemented to calculate μ_{10}.

6.2.1 Results on HLR

HLR	512 bit					1024 bit				
	RSAEKE	ZHU	GONG	OKE	GUAP	RSAEKE	ZHU	GONG	OKE	GUAP
	47	121	31	127	31	94	780	234	752	89
	16	126	31	99	15	78	812	219	778	120
	16	102	32	127	18	93	765	219	768	104
	15	126	31	97	15	78	780	203	782	73
	16	111	31	127	15	93	794	234	795	79
	16	111	32	127	0	78	761	203	820	120
	15	126	31	142	15	94	764	219	805	73
	0	141	47	123	0	109	760	203	787	89
	16	111	31	110	18	93	748	250	778	135
	16	142	16	112	15	78	789	218	773	89
	16	107	16	127	18	78	764	188	763	112
	15	94	31	140	0	109	748	203	783	73
	16	95	32	102	15	93	795	218	771	135
	15	109	47	95	31	93	795	188	751	73
	16	126	31	115	15	93	786	218	772	73
	16	110	31	127	31	79	789	235	761	89
	0	140	32	122	15	63	769	219	748	104
	16	109	31	120	16	94	821	234	779	120
	16	125	31	140	18	63	794	218	773	104
	16	110	16	120	15	78	760	235	783	120
Average (msc)	15.8	117.1	30.6	120	15.8	86.6	778.7	217.9	776.1	98.7

Table 2. Simulation Results on HLR

The results on HLR stated in Table 2, RSA-EKE resulted to perform best on HLR, where GUAP comes second with a small difference; Gong et al's protocol is about two times slower than GUAP.

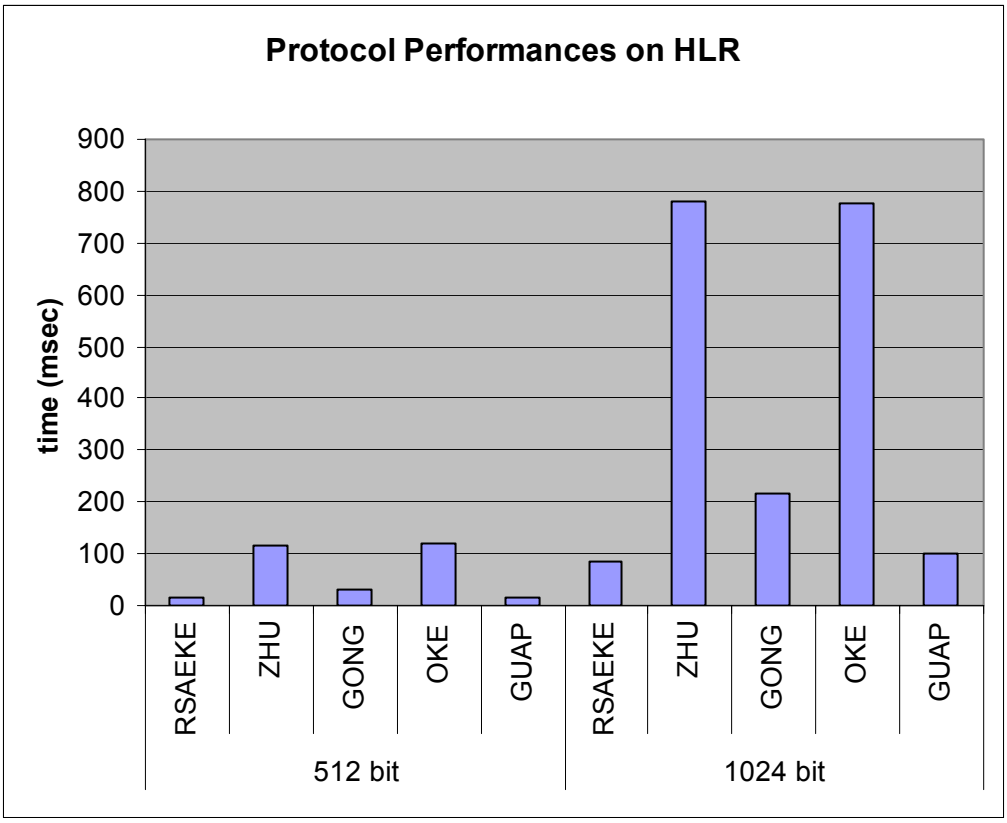

Figure 15. Average Simulation Results on HLR

Zhu et al's protocol and OKE take more computational time than other protocols because of their precautions against e-residue attack, on both protocols server does extra RSA-decryption in order to prove the correctness of the sent RSA key pair. Gong et al's protocol takes twice computation than GUAP on HLR, because in GUAP VLR is a non-user entity so communications between HLR and VLR do not need to use assymetric cryptography. This make HLR to compute one less RSA decryption in GUAP.

6.2.2 Results on VLR

EKE, OKE, and Zhu et al's protocol designed for two parties, only Gong et al's protocol and GUAP have run on VLR. In GUAP VLR is a non-user entity where it can use symmetric key operations on authentication with its device key instead of public key operations with weak secret. For this reason it is much faster than Gong et al's protocol on VLR.

VLR	512 bit		1024 bit	
	GONG	GUAP	GONG	GUAP
	234	15	235	0
	234	16	219	0
	234	0	235	0
	188	0	219	0
	250	14	250	0
	188	0	218	15
	204	0	266	0
	219	16	188	0
	204	0	234	0
	203	0	219	9
	203	0	235	0
	234	0	234	0
	219	0	203	0
	188	0	203	0
	250	0	219	10
	157	0	203	0
	234	0	234	0
	188	0	203	0
	234	14	219	0
	188	0	282	0
Average (msc)	212.7	3.8	225.9	1.7

Table 3 Simulation Results on VLR

The average results shown in Table 3 illustrate the benefit of using VLR's secret key to avoid asymmetric operations.

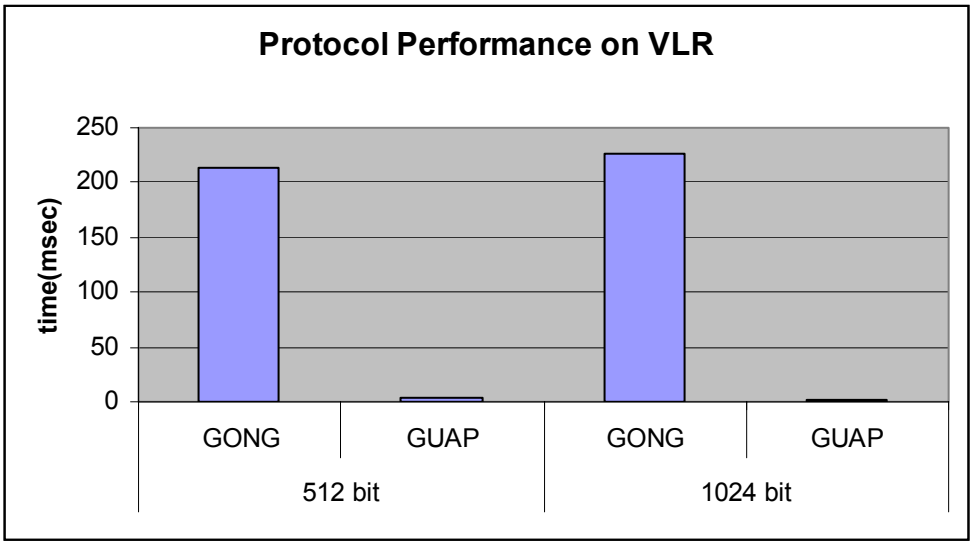

Figure 16 Average Simulation Results on VLR

The asymmetric encryption of the message in VLR makes Gong et al's protocol slower than GUAP.

6.2.3 Results on Mobile Client

Mobile Clients are more sensitive on computation time than PC's, their capacity is restricted both on memory and cpu and also usually their processors are using 16 bit architecture instead of 32 or 64 bit. As we stated above mobile clients are simulated on KToolbar, We also made simulations on Sony Ericcson's and Nokia's simulators and they gave similar results to KToolbar.

Mobile	512 bit					1024 bit				
	RSAEKE	ZHU	GONG	OKE	GUAP	RSAEKE	ZHU	GONG	OKE	GUAP
	516	2053	109	2048	114	469	2010	246	2097	211
	500	1940	14	2017	112	610	2081	270	2161	282
	438	1787	132	2095	114	610	1996	258	2329	197
	579	1996	124	1748	113	610	2081	233	2021	216
	532	1940	124	2121	98	656	2095	232	1892	254
	515	2010	124	2054	84	360	2025	212	2187	222
	218	1968	116	1927	113	1015	1884	233	2235	240
	673	1947	99	2213	98	657	2026	257	2214	212
	468	2039	108	1982	98	687	2307	219	2227	198
	483	2108	117	2023	113	578	2376	246	2306	212
	236	2052	108	1835	98	515	2215	223	2017	198
	766	1969	116	2169	98	375	2207	223	2253	211
	859	1969	116	2105	98	657	2194	223	2237	193
	220	1968	97	1957	113	625	2103	270	2150	193
	547	1982	124	2027	99	625	2250	235	2278	198
	578	2095	108	2159	112	359	2208	234	2388	221
	296	2011	99	2173	113	360	2292	222	2205	225
	561	2108	99	1812	98	360	2147	223	1987	205
	625	2038	133	2007	112	1015	2165	234	2242	240
	280	1834	133	1712	105	718	2206	259	2165	196
Average (msc)	494.5	1990.7	115.5	2009	105.2	593.05	2143.4	237.6	2180	216.2

Table 4 Simulation Results on Mobile Client

GUAP and Gong et al's protocol resulted in closer time intervals; the RSA key generation per session makes RSA-EKE slower than GUAP and Gong et al's. Protected OKE and Zhu et al's protocol performed worse because of their precautions against e-residue attack.

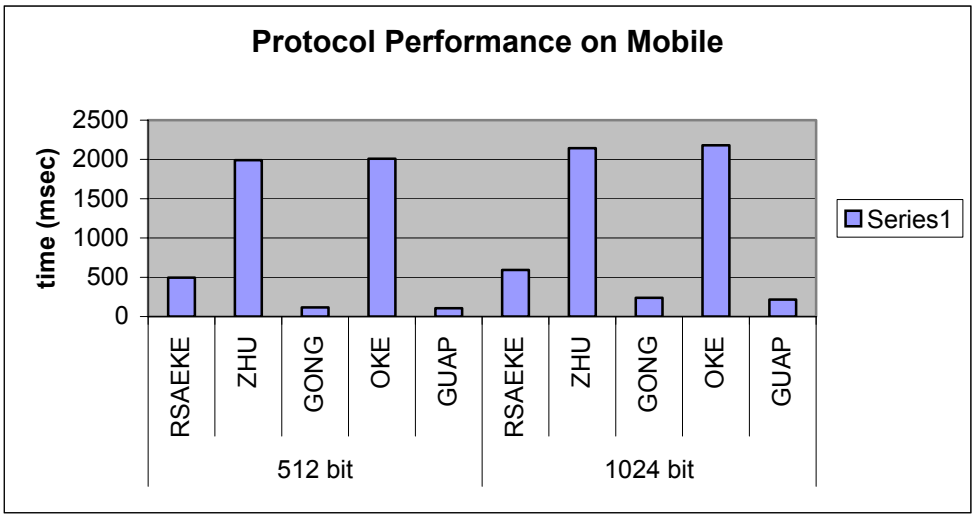

Figure 17 Average Simulation Results on Mobile Client: GUAP is the best solution to strong user authentication problem in cellular phone systems. Zhu et al's protocol and OKE resolves the per session key generation problem, however their precautions against e-residue attack (OKE's solution is still weak) make them unsuitable.

The simulation results show that the GUAP computations can be carried out efficiently in a reasonable time by all the parties, either with 512- or 1024-bit RSA. The load on the VLR is particularly low, as a result of the design decision to use symmetric key encryption between the VLR and HLR.

Zhu et al.'s protocol seems to be significantly slower. However it must be noted that this protocol was designed for a somewhat more restricted setting where the mobile device does not have a priorly established trust with the server and cannot have the server's public key installed securely beforehand. Nevertheless, we included it in the simulation experiments due to its significance as the only strong password protocol designed specifically for constrained mobile devices.

7. Conclusion

GSM is widely used over the world. If user authentication becomes possible for mobile users, everybody will be able to reach their accounts without their SIM card. People can redirect their calls through Internet, or reach their accounts through anybody's phone only by entering their username and password. We have presented a strong user authentication protocol for GSM that permits user authentication to the standard. Our protocol is inspired by strong authentication protocols for weak secrets [1, 5]. Our main goal is to break the dependency on SIM cards for authentication in GSM and to make the standard more flexible for users. The design takes into consideration the computational restrictions of the mobile subscribers. It also enables authentication of VLR by both mobile subscriber and HLR. Besides; easy, fast, and trusted key disabling can be obtained by a minor extension to our protocol.

As a final remark we would like to note that our protocol, although designed for GSM, is not particularly specific to GSM and can easily be adapted to any other mobile protocol where a user device authenticates itself to its home server via a local base station.

REFERENCES

[1] S.M. Bellovin and M. Meritt, “Encrypted Key Exchange: Password based protocols secureagainst dictionary attacks”, in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, May, 1992, pp.72-84.

[2] S.M. Bellovin and M. Meritt, “Augmented Encrypted Key Exchange: A password based protocol secure against dictionary attacks and password file compromise”, Technical Report, *AT&T Bell Laboratories*, 1994.

[3] ETSI/TC Recommendation GSM 03.20, *Security Related Network Function*, version 3.3.2, January1991.

[4] W.Ford, B.S.Kaliski, “Server assisted generation of a strong secret from a password”, WETICE2000, MD, USA

[5] L. Gong, T.M.A. Lomas, R.M. Needham, J.H. Saltzer, “Protecting poorly chosen secrets from guessing attacks”, *IEEE Journal on Selected Areas in Communication*, Vol.11, No:5, June 1993, pp. 648-656.

[6] http://www.gsmworld.com/news/statistics/index.shtml

[7] M.Van der Heijden, M.Taylor, “Understanding WAP, Wireless Applications, Devices, and Services”, 2000 Artech House Publishers

[8] Hung-Yu, Lein Harn and Vijay Kumar, “Authentication protocols in wireless communications”, *ICAUTO’ 95*.

[9] D. Jablon, “Strong password only authenticated key exchange”, *ACM Computer Communications Review*, October 1996.

[10] T.Karrygiannis, L.Owens “Wireless Network Security, 802.11, Bluetooth and Handheld Devices”, 2002, NIST Special Publication

[11] KToolbar, A toolkit for J2ME, http://java.sun.com/j2me

[12] Lightweight Crypto API, Bouncy Castle, http://www.bouncycastle.org

[13] L.Lough, “A Taxonomy of Computer Attacks With Applications to Wireless Networks”, 2001, PhD Dissertation, Virginia Polytechnic Institute

[14] S. Lucks, “Open Key Exchange: How to defeat dictionary attacks without encrypting public keys”, *Proc. of the Security Protocols Workshop*, LNCS 1361. Springer-Verlag, Berlin, 1997.

[15] P. MacKenzie and M. K. Reiter. “Networked cryptographic devices resilient to capture”, *International Journal of Information Security*, November 2003.

[16] P.MacKenzie, S.Patel, R.Swaminatham, “Password authenticated key exchange based on RSA”, In proc. ASIACRYPT 2000, pages 599-613, 2000

[17] Palm, “Securing the handhelde environment –An enterprise Perspective”,2001 Palm Inc.

[18] R. Perlman and C. Kaufman, “PDM: A new strong password based protocol”, *Proceedings of the 10th USENIX Security Symposium*, August 2001.

[19] M. Rahnema, “Overview of the GSM system and protocol architecture”, *IEEE Communications Magazine* pp. 92-100, April 1993.

[20] Greg Rose, “Authentication and Security in Mobile Phones”, 1998, Qualcomm Australia

[21] R.Steele, C.C.Lee, P.Gould, “GSM cdmaOne and 3G systems”, Wiley, 2001

[22] Wayne A.Jansen, “Authenticating User on Hadnheld Devices”, 2003, NIST

[23] T. Wu, “Secure remote password protocol”, *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pp.97-111, 1998

[24] F.Zhu, D.S. Wong, A.H. Chan, R.Ye., “Password Authenticated Key Exchange based on RSA for Imbalanced Wireless Networks”, ISC 2002, Sao Paolo, Brazil.

[25] http://gsm.argospress.com