

# Over the Air Service Provisioning

Sarvar Patel

Bell Labs, Lucent Technologies  
67 Whippany Rd, Whippany, NJ 07981, USA  
sarvar@bell-labs.com

**Abstract.** Mobile users should be able to buy their handsets and then get service from any service provider without physically taking the handset to the provider's location or manually entering long keys and parameters into the handset. This capability to activate and provision the handset remotely is part of the current North American wireless standards and is referred to as 'over the air service provisioning' (OTASP). We examine current proposals and point out some of their limitations. Often the knowledge shared between the mobile user and the network is not fully specified and hence not exploited. We depart from this norm by first providing a classification of various sharing of secrets and secondly we make explicit the assumed shared knowledge and use it to construct various schemes for OTASP. We present a different OTASP scheme for each of the following assumptions: 1) availability of a land line, 2) public key of a CA in the handset, 3) weak secret shared by the mobile user and the network, and 4) secret of the mobile user which can only be verified by the network.

## 1 Introduction

In the future, more users will continue to migrate to mobile phones and the existing users of analog phones will also migrate to authenticated digital mobile phones. In order to have the mobile handset activated, first, authorizing information from the user is needed by the service provider, and then parameters like long lived keys, telephone numbers, and other information need to be securely distributed to the handset. This entire process will be referred to as service provisioning. It is important that the mobile users have their handsets provisioned in the most convenient way possible. Ideally, the best methods for provisioning of handsets should allow for:

1. Interchangeability of handsets: Users should be able to buy handsets from any vendor as done currently with wired phones.
2. Interchangeability of service providers: Users should be able to get service from any service providers in their region.
3. Spontaneity: Users should be able to activate service when they want without having to wait to receive special codes, PINs, or even provisioned handsets.
4. Remote provisioning: Users should be able to provision their handsets remotely, over the air, without having to take the phone to a special service location.

5. Convenience: The service provisioning process should not require the users to enter long parameters manually into the phones.

This does not mean that other arrangements are not possible, for example where the handset is bought with service already activated. And furthermore, service will only be provided by the service providers if authorizing information (e.g. credit card information) is given by the mobile user and verified.

### 1.1 Example OTASP Scenario

A mobile user buys a handset and now wants to get service. The user first places a wireless call to the regional service providers (one of the calls allowed by the unprovisioned handset) using its handset id number. The service operator comes on-line and requests authorizing information (e.g. credit card information) which the user provides. Then the network downloads to the phone its long-lived key, telephone number, and other parameters. Now the mobile user has service and can immediately make calls if so desired. This entire exchange happens securely, so that others cannot eavesdrop on the authorizing information nor can they successfully impersonate as the network to the user or impersonate as the user to the network.

### 1.2 North American OTASP Proposal

Currently, the North American cellular standard specifies an OTASP protocol [13] using Diffie-Hellman key agreement. First the network transfers a 512 bit prime  $p$  and a generator  $g$ . Then a Diffie Hellman key exchange occurs as the mobile and the network pick their respective random numbers  $R_M$  and  $R_N$  and calculate and exchange the exponentials  $g^{R_M} \bmod p$  and  $g^{R_N} \bmod p$ . Each raise the exponential by their secret random exponent to form  $g^{R_M R_N} \bmod p$ . The 64 least significant bits of  $g^{R_M R_N} \bmod p$  serve as the long-lived key, called the A-key.

The Diffie Hellman key exchange is secure against passive attacks. Furthermore, active attacks may be difficult to mount on the radio interface if transmission of messages need to be blocked or substituted, thus offering some “practical” security against active attacks. For our discussion, we sidestep the radio issues by assuming the existence of active attackers. Below we list some limitations of the proposal.

#### Limitations

1. The first problem with this use of the Diffie-Hellman key exchange is that it is unauthenticated and susceptible to a man-in-the-middle attack. An attacker can impersonate the network and then in turn impersonate the user to the network. This way the attacker can select and know the A-key as it relays messages between the user and the network to satisfy the authorization requirements. The man in the middle attack has to be carried on the voice channel also.

2. The parameters  $g$  and  $p$  are unverified which means that an attacker can replace them, for example, with a prime  $p'$  such that  $p' - 1$  has small factors which will allow the attacker to solve the discrete log and recover  $R_M$  from the exponential  $g^{R_M} \bmod p'$  sent by the mobile. This is another way for the attacker to discover the A-key on the mobile side while performing a man in the middle attack. A more powerful attack, which we describe next, is thwarted by a good choice made by the protocol. Lets say the attacker sends some  $g^x \bmod p$  to the network which forms  $g^{xR_N} \bmod p$ . The attacker now has to send some  $y$  to the mobile such that  $y^{R_M} \bmod p' = g^{xR_N} \bmod p$ . Since  $R_M$  has been recovered by the attacker from  $g^{R_M} \bmod p'$ , this equation may be solvable for  $y$ . If so, the attacker now knows the A-key shared by the handset and the network. This attack is more powerful than the previous one because the same A-key, known to the attacker, would be residing in the mobile and the network. Fortunately, this attack is not possible because the protocol requires the mobile to receive the networks exponential,  $g^{R_N} \bmod p$ , along with  $g$  and  $p$  before it will send its exponential  $g^{R_M} \bmod p$ .
3. There are no provisions to check for the equivalent of weak keys for Diffie Hellman. For example if the attacker substitutes 1 or -1 as the exponentials exchanged on both sides then the Diffie Hellman key will be 1 or -1 and known or easily guessable to the attacker. Note that this attack does not require the attacker to be in the middle encrypting and decrypting the transmission under different keys. Once 1 or -1 has been substituted then the attacker only has to eavesdrop. Also, importantly the same Diffie Hellman key, 1 or -1, is agreed upon by both the handset and the network with probability  $\frac{1}{2}$ . If the prime  $p$  is not a safe prime then other values are also possible; as a precaution all of them should be checked, and if the Diffie Hellman key matches any of them then the session should be aborted by the network.
4. We present another attack which is undetectable by the network, unlike the previous attack, yet the same known A-key will reside in the network and the handset. The attacker performs a Diffie Hellman key exchange with the network and agrees on a 512 bit key,  $K$ . The attacker now wants the mobile to have the same 512 bit key,  $K$ . The attacker, pretending to be the network, sends to the mobile  $(g, K + 1, K)$  instead of  $(g, p, g^{R_N} \bmod p)$ . The handset calculates  $K^{R_M} \bmod (K + 1)$  as the mobile's key. This key will equal one if  $R_M$  is even and the key will equal  $K$  if  $R_M$  is odd. Thus with probability  $\frac{1}{2}$  the attacker has forced the network and the handset to have the same 512 bit key,  $K$  and thus the same 64 bit A-key.
5. The protocol is a key agreement protocol which is normally fine, but a key distribution protocol where the network creates a well chosen and random (i.e. unpredictable) A-key and distributes it to the handset would have been preferable in this situation. This may allow the network to associate the A-key of a handset in some deterministic manner which is good for its key storage and management procedures. This can be done via the use of pseudo random functions (PRFs) which is indexed by a master key and maps handset/phone number into 64 bit A-keys [8]. Thus instead of storing thousands and perhaps millions of keys and protecting them, the network can store

just one master key and use the PRF to recover a particular A-key associated with the handset's number. This master key and the related algorithm can be performed in a protected device, thus the A-keys are never seen by anyone.

6. The size of the prime, 512 bits, may be too short.

The above attacks should not be surprising once it is decided that an unauthenticated Diffie Hellman key exchange will be used. In fact one can make a general statement that any unauthenticated key exchange will reveal the authorizing information (e.g. credit card number) to an active adversary. When a mobile requests an OTASP session, the active attacker blocks the session and, pretending to be the network, carries an OTASP conversation with the mobile user. Once the user reveals the credit card number, the attacker terminates the session and can use the credit card number to get service on other mobile phones or use it for other fraudulent purposes. Thus the security in such protocols lies in the practical difficulty of implementing active attacks. Despite its limitations, this is an interesting application of the Diffie Hellman key agreement protocol and makes service provisioning much more convenient.

### 1.3 Carroll-Frankel-Tsiounis Key Distribution Proposal

This is a key distribution proposal which has the benefit of allowing the network to randomly choose strong keys from the space of A-keys. The Carroll-Frankel-Tsiounis (CFT) proposal also uses Rabin to speed up computation as did the Beller-Chang-Yacobi [1] protocol and also assumes that each handset possesses the public key of a certificate authority (CA). However, it interestingly differs from other protocols in its extensive use of unforgeable signatures and encryptions which are semantically secure or plaintext aware. The protocol is as follows:

1. The network sends the mobile its public encryption key signed by the CA (unforgeable signatures are used here).
2. The mobile verifies the network's public encryption key and then generates a random session key SK and a random pad AP. It encrypts both the SK and AP using the network's public encryption key and sends it to the network (the encryption here is semantically secure).
3. The network recovers the SK and the AP and uses the SK to perform symmetric encryption of the A-key and the AP which it sends to the mobile (symmetric encryption here is plaintext aware).
4. The mobile verifies the AP in the decryption; the handset and the network now both possess the A-key which is used to derive the voice encryption key and set up an encrypted voice channel.
5. At this point the operator requests authorizing information (e.g. credit card information) from the user. If the user furnishes the information then the user has been authenticated to the network and service will be provided in the future.

We make some cautionary observations that the CFT protocol should not be viewed as a "solution to all problems," but the components that surround it should also be designed within the correct security model. We provide specific examples below:

The CFT protocol assumes that a unique A-key is generated for every OTASP attempt by the handset. If the A-key generation process does not guarantee this then the overall protocol will be insecure; as an example the same A-key may be generated for the same handset/phone number combination which would allow this attack: a handset uses its serial number and phone number (if assigned) to access the network for OTASP. At this point the attacker blocks the access. Instead the attacker picks a random session key SK and a random pad AP and sends them to the network using the blocked handset's serial/phone number. The network responds with the encrypted A-key which the attacker retrieves and aborts the connection. Now the attacker is in possession of the A-key for that handset. If the legitimate handset again accesses the network with its own session key and pad, the network will again transport the same A-key to the handset encrypting it with the session key. Now the handset will have the A-key and the user, on the encrypted voice channel, will give authorizing information thus successfully completing service provisioning. Unfortunately, the attacker already has the A-key and he also can use it later to make fraudulent calls. Thus in the CFT model, one should not directly use a PRF to associate an A-key to a handset/phone number without guaranteeing uniqueness at every OTASP attempt. Later, we will show that it is not necessary to have this restriction in a key distribution protocol.

Secondly, a mild form of denial of service attack is possible if the handset/phone numbers are not used in the cryptographic operations of the CFT protocol or the later verification stages. Actually, there are two forms of denial of service attacks. In the first one, the customer's credit card is not charged and OTASP is unsuccessful, while in the second one, the customer's credit card is charged and OTASP is unsuccessful. There is little that can be done to prevent the first form of the attack in any protocol, but we would like to assure the customers that if their credit cards are charged then OTASP will be successful and their service will be activated. An attacker can perform the second form of attack by substituting other numbers in place of the handset's true id number and user's phone number throughout the protocol. The protocol will be completed and credit cards charged, but the network will not have activated the true handset/phone number. Verification steps done after the CFT protocol can also be satisfied as long as they do not use handset/phone numbers as part of the cryptographic operations. Thus later attempt by the user to access the system will be rejected. This attack is possible because the handset id number used in communication is not part of the public key encryption of the SK and the AP sent by the mobile to the network. If it was then the network can know that no attacker could have substituted another false handset/phone number. Fortunately, the current North American does perform cryptographic operations using the handset/phone numbers in the verification steps following the A-key

agreement. So the attack would be detectable at this time, and the credit cards should not be charged until after verification steps.

Finally the CFT protocol tries to minimize the impact due to lack of strong random sources and points out that if independent sources of randomness are used then a weak source for the SK is not problematic as long as the other AP is strongly random. We caution that this is not true if the CFT protocol is embedded in the current North American standard because candidates for the SK can be used to decrypt the symmetric encryption and give candidates for the A-key. These candidates can be further verified because the A-key is used to derive another session key called the SSD which is further used to answer challenges. An attacker can see if a candidate A-key results in the same response to the challenge as actually seen in a session. If not then the next candidate is tried until the true A-key is recovered. However, these off-line verifications of SK guesses would not be possible if the challenge/response protocol following the CFT protocol was replaced by a more secure authentication protocol (i.e. zero knowledge) which does not reveal information about the A-key. Nevertheless, the weak source for SK must have some minimum strength so that on-line verifications of A-key are not practical. Assume an attacker has good guesses for SK and hence, guesses for A-key. If these guesses are few then the attacker can use each A-key guess to establish a session (e.g. call origination). The attacker will be unsuccessful on all tries except the one with the true A-key.

## 2 Classification of Shared Knowledge

Different OTASP schemes are possible depending upon what shared knowledge is possessed by the mobile user/handset and the network. At one extreme, one can assume that the mobile handset and the network each have a public key and both keys are known to the handset and network. Similarly, for the symmetric cryptography case, we can assume that the mobile handset and the network both share a strong secret (64 bits or more). Next we can assume that the mobile handset possesses the public key of a CA and thus indirectly has knowledge of the network's public key. At the next level we can assume that the mobile user and the network only share a weak secret. Finally, the weakest assumption is that the network can only verify a secret relayed to it by the mobile user. That is the network does not even share a secret with the user, but can only verify it.

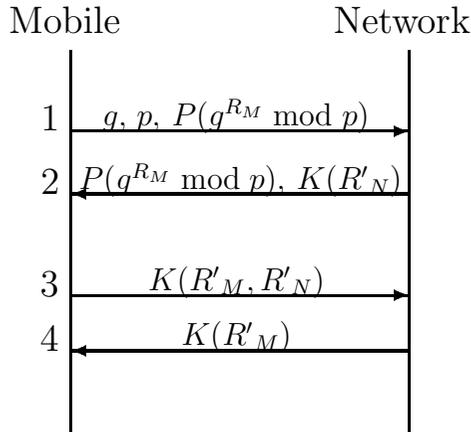
Orthogonal to this is the availability of a secure channel (land line phone) which can aid in OTASP. Although, we describe the various schemes with respect to the wireless environment, they can be used in other insecure environments (e.g. internet) where there is a need to remotely provision. For different situations, different assumptions may be valid. In some environment there may be an agreement on the certificate authority and hence on the root key to be pre-stored in every device. However, for other environments no easy agreement on a CA and its procedures may be possible and hence, the other non-CA based schemes need to be used.

### 3 Secure Password Scheme

We want to review in this section methods to use weak secrets (e.g. passwords) for authentication and key agreement. These methods will be used repeatedly. Standard protocols used in symmetric key authentication and key agreement are susceptible to off-line dictionary attacks. A typical protocol consists of a challenge  $R$  sent from user A to user B. User B responds with a function  $f_P(R)$ , using the challenge  $R$  and the password  $P$ . An eavesdropper hearing the pair  $R$  and  $f_P(R)$  can use this information to verify guesses. Users do not pick passwords randomly from the entire  $2^n$  possible space, but instead use memorable and meaningful passwords. Quite often their passwords are picked from names and words in dictionaries. So an attacker can perform off-line dictionary attack by picking each word,  $P'$ , in the dictionary and forming  $f_{P'}(R)$  and seeing if it matches  $f_P(R)$ . If it does not then the attacker tries the next guess until it finds a match which reveals the password  $P$ . No matter how complicated the protocol, if it only uses symmetric cryptography then it will be susceptible to a variation of the off-line dictionary attack.

#### 3.1 Review of Secure Password Protocols

There has been some advance towards password protocols resistant to off-line dictionary attacks [10], [3], and [9]. Lomas et.al [10], were the first to propose a 3 party protocol using passwords which were protected against dictionary attacks. Bellare and Merritt [3] made similar protocols called Encrypted Key Exchange (EKE) for two party authentication and key exchange using passwords and still had protection against dictionary attacks.



**Fig. 1.** The Diffie Hellman Encrypted Key Exchange (DH-EKE)

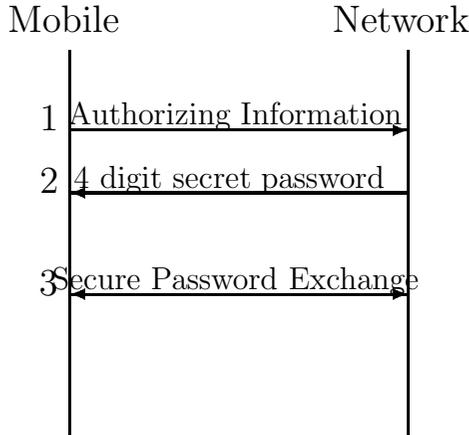
[11] showed that information leakage is possible in EKE which allows one to retrieve the passwords after hearing some exchanges. Furthermore [12] presented attacks against variations of EKE and Gong et.al protocols where varying or unverified parameters were used. The fixing or verifications of parameters seems necessary. RSA parameters are unverifiable and hence must be fixed. However, Elgamal and DH (Diffie Hellman) parameters like  $g$  and  $p$  are verifiable and hence can either be fixed or changing but verified during each exchange.

We now describe Bellovin and Merritt's secure password protocol, called Encrypted Key Exchange(EKE) based on the Diffie-Hellman key exchange (DH-EKE). The mobile and the network (see figure 1) have agreed upon a  $g$  and a  $p$ , here we assume that the  $g$  and the  $p$  are fixed. The mobile picks a random number  $R_M$  and calculates  $g^{R_M} \bmod p$ . The mobile further encrypts the value using the password  $P$  to get  $P(g^{R_M} \bmod p)$  and sends it to the network. The network similarly calculates and sends  $P(g^{R_N} \bmod p)$  to the mobile. Both decrypt the messages and exponentiate with their respective secret random exponents to form the Diffie Hellman key,  $g^{R_M R_N} \bmod p$  which serves as the session key. The session key is then authenticated by random challenges to prevent replay and other attacks. Since the Diffie Hellman exponential  $g^{R_M}$  and  $g^{R_N}$  are encrypted using the password  $P$ , a man-in-the middle attack is not possible, furthermore, the passwords prove authentication because without knowing the passwords there would not be an agreement on the Diffie-Hellman keys. How does this stop the dictionary attacks? Well, because guessing the password  $P'$  allows one to recover the guesses for the exponentials  $g^{R_M'}$  and  $g^{R_N'}$ , but the attacker cannot form the Diffie Hellman key  $g^{R_M' R_N'}$  and verify the guess, hence off-line guesses from the dictionary cannot be verified.

Encrypting  $g^R \bmod p$  with a symmetric cipher can leak information, and Bellovin and Merritt propose a random padding method, however, this also leaks information allowing an attacker to recover the password. Some countermeasures are available [3], [11].

## 4 Secure Channel (Land Line) Available

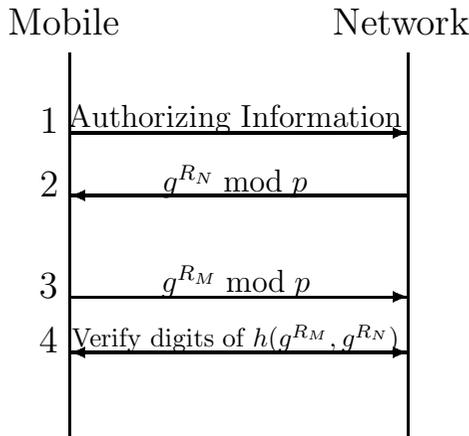
Assuming a land line connection is available we present two different schemes for OTASP which are secure against man in the middle attacks. The first method of performing a secure over the air key exchange, uses a secure password protocol (see Figure 2). First the user contacts the operator over a land phone line (First two lines in the figure refer to a land line connection and the third line refers to a wireless connection). The land phone line serves as our authenticated and private channel. The operator asks questions to verify that the user is authorized. Then the user is instructed to power on the handset and enter a 4 digit number provided by the operator. The network then uses this 4 digit number as the password to perform a secure password Diffie-Hellman key exchange as described in the previous section. Once a temporary DH key has been agreed then the channel can be encrypted and the messages authenticated using symmetric techniques as used for executing a secure session. This way the A-key and



**Fig. 2.** Over the Air Key Exchange - Using Land Lines with a Password Protocol

other parameters can be securely and privately downloaded to the handset. As an example, the telephone number can be privately downloaded and thus the anonymity of the user can be kept, including other parameters.

The second method of doing a secure over the air key exchange is to have the mobile handset display a string and have the user read it to the operator, this way the man in the middle is avoided. The protocol is described in figure 3 where the first and last passes occur with the operator on the land lines while the 2nd and 3rd occur on the wireless connection between the handset and the network. After the operator has authorizing information from the mobile user, the mobile

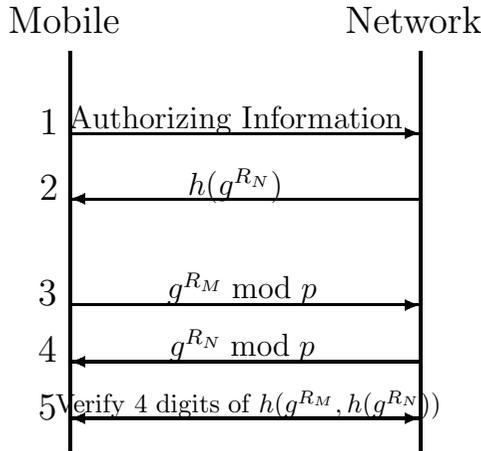


**Fig. 3.** An Over the Air Key Exchange Susceptible to Birthday Attacks

is turned on and in steps 2 and 3 the mobile and the network send  $g^{R_M}$ , and  $g^{R_N}$  to each other. The operator and the user verbally verify that some of the digits of  $h(g^{R_M}, g^{R_N})$  are the same, where  $h$  is a cryptographic hash. If verified then the network and mobile use  $h(g^{R_M R_N})$  as the A-key or else the network can initiate security violation procedures. This is good not just for wireless, but for any kind of remote update of the parameter with human verification.

There is a birthday attack which one has to be careful about. Lets say that we decide that 64 bits will be read back by the mobile user to the operator. It would seem that the attacker would have to try  $2^{64}$  different values so that the hash of those values  $h(g^{R_M}, g^{R_{N'}})$  and  $h(g^{R_{M'}}, g^{R_N})$  are the same, a necessary condition to carry the man-in-the-middle attack. So the attacker tries all the values for  $R_{N'}$  and  $R_{M'}$  such that their hashes are the same. Although naive counting would suggest that about  $2^{64}$  values need to be tried to get a significant chance of collision, we can by birthday arguments show that about  $2^{32}$  values need to be tried to find a collision with probability near  $1/2$ . To slow things down one can put  $g^{R_M R_N}$  as part of the hash:  $h(g^{R_M}, g^{R_N}, g^{R_M R_N})$ . This means the attacker has to do exponentiation along with hashes which makes things slower. If the user is asked to read back alphabetic characters then about 14 letters are needed to cover 64 bits. If alphanumeric are used then about 12 are enough. However  $2^{32}$  complexity for an attack even in real time may not be enough security and if  $2^{64}$  complexity is required then 128 bits must be read back by the user which is about 24 alphanumeric. There is a simpler method of making the protocol in Figure 3 resistant to the birthday attack by introducing a restriction on the sequence of the exchange of the Diffie Hellman exponentials. In particular, if we insist that the mobile will not send its  $g^{R_M}$  until it receives  $g^{R_N}$  from the network then the birthday attack is foiled. The man in the middle attacker was previously able to see both the exponentials  $g^{R_M}$  and  $g^{R_N}$  and thus it was able to exploit the birthday attack. Now the attacker has to commit an exponential to the mobile before it will see the mobile's exponential  $g^{R_M}$  thus reducing one degree of freedom for the attacker.

We present another version of the protocol with one more round which is not susceptible to the birthday attack, and furthermore is resistant to searches for consistent exponentials by a man in the middle attacker. Thus verification of a smaller string is sufficient. The protocol is described in figure Figure 4 where the first and the final step are over the land voice link while the middle three steps occur over the air link. On the wireless link, first the network sends the hash of its exponential, the user then sends its exponential,  $g^{R_M}$ , and finally the network sends its exponential  $g^{R_N}$ . The user first verifies the hash of the exponential sent by the network. Then both the user and the network calculate  $h(g^{R_M}, h(g^{R_N}))$  and verbally verify its first 4 digits. A man-in-the-middle attacker cannot use birthday type attacks or do searches for consistent exponentials because as a network he has to commit to the exponential he is using (via the hash) before he sees the users exponential. Similarly, the attacker as a user has to commit to the exponential before the value of the networks exponential, associated with the hash, is revealed. Thus we need to verify a much smaller string (e.g. 4 digits).



**Fig. 4.** An Over the Air Key Exchange Resistant to Birthday Attacks

There is an existing protocol, called VP1 [4] which is related to this protocol, but it is used for agreeing on a session key whereas we use our protocol to update parameters via an authenticated link. Secondly, VP1 is a 4 round protocol whereas we are a 3 round protocol on the wireless link. VP1 first has both the user and the network hash their exponentials and exchange them. Then the actual exponentials are exchanged. Finally, a hash of the function of the exponentials is calculated and 6 digits are verified by voice. Our protocol is more efficient in terms of the number of rounds and the messages exchanged. Since VP1 does not use an authenticated link, it ultimately does not protect against man-in-middle attacks, unless the end parties know each other’s voice. Our method can be used for other environments. Also other variations can be built using different public key schemes.

The safe prime  $p$  and the generator  $g$  were assumed to be fixed and prestored in the handset. However, if that is not the case then the attacker can replace  $g$  and  $p$  with  $g'$  and  $p'$  which may allow it to calculate the discrete logarithm efficiently. If  $g$  and  $p$  are also sent over the air then they could also be used as part of the hash calculation,  $h(g, p, g^{R_M}, g^{R_N})$  in order to detect the substitution of  $g$  and  $p$  by the attacker.

#### 4.1 Man-in-the-Middle on Voice Channel

Although we have described the last two protocols as requiring a land line voice link one can execute the protocols without it. Thus a handset can be used to make the voice call itself to the network operator and a Diffie Hellman performed on the control channel. No man in the middle attack is possible on the control channel as described above. However, a man-in-the-middle attack is possible if it is also performed on the voice channel. First the attacker pretends to be the

network to the user and gets the authorizing information. Secondly the attacker pretends to be the user to the network. If this is happening in real time then all the information from one conversation is relayed to the other (assume there are two attackers working in concert). Then on the control channel the man-in-the-middle attack can proceed because when it comes time to verify the digits on the voice link, the two attackers will say the different digits to the network and the user respectively. Although we have described this as happening in real time, it could happen that the attacker gets authorizing information from the user and then at a later time calls the network and gives it the information to get the A-key and other parameters.

## 5 Mobile and Network Share Strong Keys

If the handset and the network already share strong secrets either in form of each other's public keys or strong secret keys, then well known protocols can be used to exchange or update parameters. [1] is an example of a protocol using public keys to establish session keys. [2] is an example of secure protocols used with symmetric cryptography to establish session keys. The session keys can be used to encrypt a session and authorizing information can be provided to start service and the parameters can be provisioned into the handset. We are trying to get strong secrets agreed upon by both sides, but since we don't possess them initially we will try to bootstrap from weak secrets to strong secrets.

## 6 Mobile Has CA's Public Key

Manufacturers can install a CA's public key in all the phones. This is much easier than installing a unique public key for each handset. The assumption is the one used in the CFT protocol and also used in SSL. Unlike the CFT protocol, we will use the handset id as part of our encryption thus blocking the denial of service attack. Furthermore, we want our protocol to allow a network to be able to pick a well chosen A-key for a specific handset/phone number and try to distribute that A-key to the handset repeatedly. The CFT protocol was not able to do this because it revealed its hand before it was necessary. That is, it revealed the A-key before authorization was given on the voice channel. In fact our protocol does not require a voice connection to transfer the authorizing (e.g. credit card) information from the mobile user to the network. The user can be prompted on the handset and the user can enter the information on the handset. We reveal the A-key only if the authorization step has been successfully completed. Here are the steps of our protocol:

1. The handset requests OTASP and identifies itself via a handset id number or a telephone number if already assigned.
2. The network sends its public key, signed by the CA, to the handset (unforgeable signatures).

3. The mobile generates a random session key SK and encrypts the handset id and the session key SK with the network's public key and sends it to the network (using probabilistic encryption [7] or semantically secure encryption).
4. The network decrypts and then uses session key SK to initiate a authenticated and encrypted session. Voice and messages are both encrypted and message authentication is provided using a message authentication key and a encryption key derived from the session key, SK.
5. The user provides the credit card information to the network.
6. If the credit card information is verified then the network distributes the A-key to the handset.

Note that unlike the CFT protocol, the entire OTASP has been encrypted and authenticated using the session key derived from the SK rather than the session key derived from the A-key.

## 7 Mobile User and Network Share Weak Secret

It may be that the user and the service provider have had a previous contact and now share some personal information about the user. The user might have already interacted with the service provider or the user may be an existing land line customer of the service provider and now wants to get mobile service. It is not clear, what the personal information is that the network and the user share. Is it the mothers maiden name? The last 4 digits of the social security number (SS#) or zipcode? Obviously there must be some information that the operator has which it uses to authenticate the user on the voice link. Assume its the last 4 digits of the SS#. If so then we can perform a secure password protocol using the last 4 digits of the SS#. First the user enters the last 4 digits of the SS# into the handset, and then using the handset itself a secure password protocol is executed and the A-key and other parameters are updated in the handset. Note all this happens without a voice call being placed either on the land line or on the handset. In a sense, a temporary password (last 4 digits of SS#) is used and then other parameters are updated. The protocol is the same as the one in Figure 3 except all the communications take place over the wireless phone. There is no need for a real time operator to be involved, although it is not precluded from the protocol.

Perhaps credit card information can serve as the shared secret. Since the network operators have access to credit card information about all users, this information can be used as a shared secret. If an operator using the name of a person and type of card (e.g. citibank visa), can know the credit card information, then that can serve as the shared secret. This may not be possible if the operator can only verify the credit card number, but does not know the credit card number from just the name of the user. When the user contacts the network operator, the operator will ask for the user's name and type of credit card. Then the user is prompted to enter the card number which will be used as the weak shared secret to initiate a secure password protocol. If successful then the A-key

and other parameters can be distributed to the handset over an encrypted and authenticated session.

## 8 Network Can Verify Mobile User's Secret

In the worst case there is no shared secret, however, the mobile user has a secret (name + credit card number + expiration date) which the network can verify. The network does not know this information ahead of time, but can only verify it. Now we cannot use the previous techniques and hence come full circle to using something very similar to the current North American OTASP proposal. There does not seem to be any cryptographic techniques which will protect against active attackers in this situation.

We will also perform an unauthenticated Diffie-Hellman key exchange or another unauthenticated public key (e.g. Rabin) based key distribution. However, we will do it in such a way that performing a man in the middle attack is very difficult, involving service denial to much of the service region for extended periods of time. We do this by disguising an OTASP key exchange as a normal system access (e.g. call origination) followed by an encrypted and authenticated session. In order to do this, a random ID/telephone number for the handset should be used to make a call origination. The network, when it sees the random ID/telephone number will know that this is not a legitimate number. The network knows that this could happen either because there was error in transmission or some one is trying to initiate OTASP. The network then continues to pretend its a normal call, and sends random bits to the handset and the handset also sends random bits to the network. However the first 1024 bits of the disguised call can be exponentials  $g^{R_M} \bmod p$  and  $g^{R_N} \bmod p$ . The key is derived and used to encrypt the rest of the session after some predetermined time, say after 10 seconds of data. Then the call should be placed to the operator in the network and the mobile user should relay the secret credit card information which the network will verify. If the information is verified then A-key can be transported to the handset along with other parameters.

If the exchange is disguised well then the only way for the attacker to act as man in the middle is to try to do so with most calls which are going on, hoping that it will find one that is truly an OTASP call. To have any significant probability of finding such calls it will have to be blocking most calls because an OTASP call is a rare call, once or few times in the lifetime of a phone. A call origination on the other hand is very frequent, thus the cost of the attack is expensive. So if such kind of blocking or denial of service occurs then it should become easier to find the attacker and becomes all the more important to find the source and put an end to the blocking. The security of such a protocol is not cryptographic but practical. We make the practical assumption that the adversary is not so powerful as to block most calls and still go undetected.

The security of this strengthened unauthenticated Diffie Hellman is analogous to the strength of passwords schemes in withstanding on-line attacks. An attacker guessing at the correct password in one session has a low probability of

guessing it correctly, but guessing over thousands of sessions the attacker has a high probability of recovering a password for some user. Similarly, an attacker performing a man in the middle attack on a session has a low probability of success, but over thousands of sessions the probability of success is high.

The Diffie Hellman exchange is only strengthened if the OTASP call is indistinguishable from a normal call origination and this is very tricky to guarantee. For example, if  $g$  and  $p$  were sent during the OTASP call then the call is distinguishable because an attacker can check if  $p$  is a prime. If not then the attacker knows that this is a normal call origination and there is no need to perform a man in the middle; we had assumed that  $g$  and  $p$  are fixed and known. If we do want to send  $g$  and  $p$  as part of the session then [11] outlines some methods to do this safely. A general method of sending primes is to send the random string used in picking a prime rather than sending the actual prime. At the other end, the same random string is used to pick the same prime or a safe prime. To transfer  $g$ , a random string is sent to the other party who checks if the string is a generator of  $p$ . If so then it is used as the  $g$  else the increment is checked until a generator is found.

## 9 Conclusion

We started by examining the proposals for OTASP and their limitations. We then provided over the air service provisioning methods under various assumptions about knowledge shared between the mobile user/handset and the network. We started from strong assumptions like the knowledge of strong keys and moved to the weakest assumption, that the mobile and network do not share any secrets but if the mobile relays its secret the network can verify it. For this weak assumption we could only provide a practically secure scheme assuming a limited adversary. If agreement on public keys is desired, we can further have the handset and network generate their respective private and public keys and exchange them. Thus we are able to bootstrap from a small secret to a strong public key. The table (Figure 5) organizes the various protocols according to the different assumptions. The left column of the table lists the various assumptions about the secrets shared between the network and the handset. The top row of the table lists the various 4 assumptions about availability of a land line, or the availability of pre-stored and public constants like the CA's public encryption key or the  $g$  and  $p$ , or the possibility that nothing is pre-stored.

If a strong secret is shared then a standard two party session key agreement protocol can be used for all 4 assumptions. If a weak secret is shared between the handset and the network then the schemes of section 4 can be used with the availability of land lines; if the CA's public encryption key is pre-stored in the handset then the protocol of section 6 can be used; if  $g$  and  $p$  are pre-stored then the secure password scheme can be used and even when  $g$  and  $p$  are not pre-stored the secure password scheme can be used as long as the  $g$  and  $p$  are verified. Finally, the weakest assumptions on secrets is that the mobile user has a secret, but the network does not share it and can only verify it when presented by the

	Land line available	CA's public key in handset	$g, p$ in handset	nothing pre-stored in handset
<b>Strong secret</b>	session key agreement	session key agreement	session key agreement	session key agreement
<b>weak secret</b>	secure password	CA scheme	secure password	secure password ( $g, p$ verify)
<b>Verifiable secret, but unshared</b>	secure password or verify hash	CA scheme	strengthened DH	strengthened DH with transfer $g, p$

**Fig. 5.** Summary Table

mobile user. In this case, the availability of land lines means that the protocols from section 4 can be used; if the CA's public encryption key is pre-stored then the protocol of section 6 can be used here also; if  $g$  and  $p$  are pre-stored then one has to resort to the strengthened version (section 8) of the unauthenticated Diffie Hellman key exchange which makes the OTASP call indistinguishable from a normal call; even if  $g$  and  $p$  are not pre-stored then one can use the protocol in section 8 as long as  $g$  and  $p$  are carefully transferred as outlined in section 8.

## References

1. M. Beller, L. Chang, and Y. Yacobi, Privacy and authentication on a portable communication system, *IEEE J. Select. Areas Commun.*, 11: 821-829, 1993.
2. M. Bellare, and P. Rogaway, Entity authentication and key distribution, *Advances in Cryptology - Crypto*, 1993.
3. S. Bellovin, and M. Merritt, Encrypted key exchange: password-based protocols secure against dictionary attacks, *IEEE computer Society symposium on research in security and privacy*, 72-84 May 1992.
4. E. Blossom, The VP1 Protocol for Voice Privacy Devices, December 1996.
5. C. Carroll, Y. Frankel, and Y. Tsiounis, Efficient key distribution for slow computing devices: Achieving fast over the air activation for wireless systems, *IEEE symposium on security and privacy*, May, 1998.
6. EIA/TIA, Cellular Radio Telecomm. Intersystem Operations IS-41C 1994.
7. S. Goldwasser, and A. Micali, Probabilistic encryption, *Journal of Computer and Systems Science*, 28: 270-299, 1984.
8. O. Goldreich, S. Goldwasser, and A. Micali, On the cryptographic applications of random functions, *Advances in Cryptology - Crypto*, 1984.
9. L. Gong, T. Lomas, R. Needham, J. Saltzer, Protecting poorly chosen secrets from guessing attacks, *IEEE J. Sel. Areas Commun.*, 11: 648-656, 1993.
10. T. Lomas, L. Gong, J. Saltzer, R. Needham, Reducing Risks from Poorly Chosen Keys, *ACM Operating Systems Review*, 23(5): 14-18, Dec. 1989.
11. S. Patel, Information Leakage in Encrypted Key Exchange, *Proceedings of DIMACS workshop on Network Threats*, 38: 33-40, December 1996.
12. S. Patel, Number theoretic attacks on secure password schemes, *IEEE symposium on security and privacy*, 236-247 May 1997.
13. TIA, IS-41-C enhancements for Over-The-Air-Service Provisioning, *TR45.2 Working Group*, December 1996.