

Key Recovery in Third Generation Wireless Communication Systems^{*}

Juanma González Nieto¹, DongGook Park^{1,2}, Colin Boyd¹, and Ed Dawson¹

¹ Queensland University of Technology, Information Security Research Centre,
GPO Box 2434, Brisbane, Queensland 4001, Australia

² Korea Telecom, Access Network Laboratory,
17 WooMyeon-Dong, SeoCho-Gu, 137-792, Seoul, Korea

Abstract. Future mobile communications networks, so called third generation systems, may need end-to-end security in some applications involving value-added services such as providing secure communications between a user and a bank in electronic commerce. The provision of end-to-end security may require mechanisms for key recovery. In this paper we identify security flaws with a previous published protocol for key recovery in such networks. A new key recovery protocol which overcomes these flaws is presented.

1 Introduction

Third generation wireless mobile communication systems will realise a new vision of telecommunication services through the provision of enhanced current generation services such as cellular, PCS and radio-paging within their unified architectures, and of a more diverse range of telecommunication services including multimedia and Internet services. Two main systems have been proposed for future generation mobile applications: the International Mobile Telecommunications-2000 (IMT-2000) [16,6] on a worldwide basis, and the Universal Mobile Telecommunications System (UMTS) [4] in the European context. UMTS, like its predecessor Global System for Mobile (GSM), seems to have influenced worldwide standardisation development for third generation systems.

UMTS and IMT-2000 will take advantage of many advanced security technologies, especially asymmetric (or public key) cryptography, at least in their fully developed stages [15]. Public key based security, with a mature public key infrastructure (PKI), will enable all the involved entities such as users, network operators (NOs), value-added service providers (VASPs) and service providers (SPs) to have full range of state-of-the art security features including:

- non-repudiation services for incontestable charging and electronic commerce.
- mutual authentication between the user and the NO/VASP without accessing the home SP on-line, thus enabling seamless roaming of the user.

^{*} This research is part of the co-operative project Security Technologies in Wireless Communications between Queensland University of Technology and Korea Telecom

All the security features depend on the successful execution of a wireless authentication and key establishment (WAKE) protocol between the user and the NO/VASP. Figure 1 shows all the entities involved in the public key based security architecture of future mobile communications as well as the interfaces over which WAKE protocols will be run.

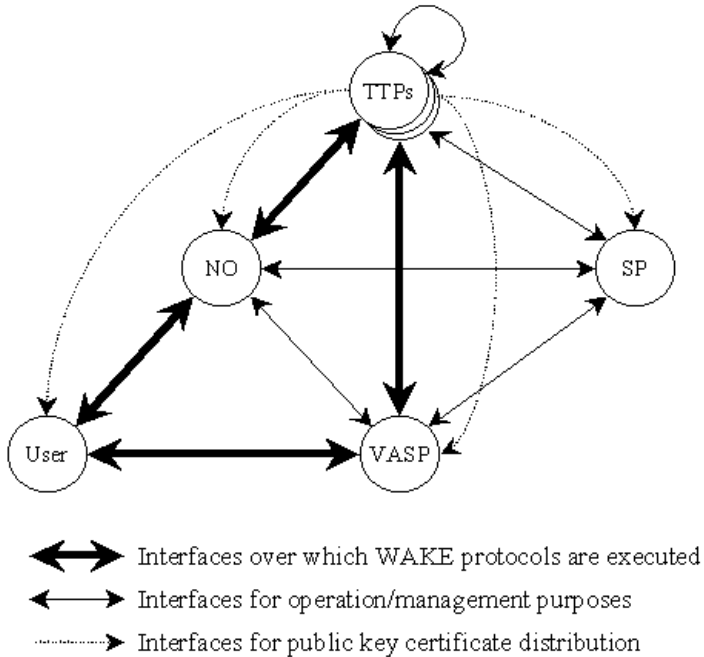


Fig. 1. Role players in future wireless mobile communications using public key infrastructure

Trusted third parties (TTPs) will serve as clearing houses with regard to all the required trust relations for WAKE protocol executions between the user and the NO/VASP, both on an on-line and off-line basis. The User-to-VASP interface is a logical interface to be established through NO networks, and has some interesting issues such as a rather strict requirement for non-repudiation services for electronic payment, and some appropriate key recovery mechanisms. The latter is the main issue of this paper.

The user and the VASP need to authenticate each other and establish a shared secret session key to encrypt the subsequent message exchange. This is achieved through execution of the relevant WAKE protocol which may or may not be the same as the WAKE protocol for the user-to-NO interface. Because VASPs, unlike NOs and SPs, are a special kind of end user, the user-to-VASP

communication can be regarded an end-to-end link, which gives rise to a problem in relation to lawful interception. The information protected with user-to-NO session keys can be accessed by law enforcement agencies (LEAs) with the aid of the relevant NO (when authorized) without using any special cryptographic mechanisms. End-to-end encryption, however, requires a special cryptographic facility, a *key recovery* mechanism, to enable a LEA to access the plaintext encrypted under an end-to-end encryption key. Key recovery, although still a matter of debate, is expected to be deployed in the UMTS security architecture to satisfy the government requirement of lawful interception [12].

In this paper, we first review a proposed key recovery protocol for the prominent UMTS WAKE protocol. This is followed by a security analysis of this protocol which identifies several vulnerabilities. A new key recovery protocol is then proposed which is not only secure from the vulnerabilities of the UMTS proposal but is also more computationally efficient. The proposed new protocol utilises a simple mechanism for verifiable encryption appropriate for use in this context.

2 Background on Key Recovery

Cryptography protects the confidentiality of information by limiting access to plaintext of encrypted data to those that possess the corresponding decryption keys. This in turn requires the deployment of key management techniques for the secure administration (generation, distribution, storage, etc) of cryptographic keys. In particular, mechanisms might be needed to allow extraordinary access to the plaintext data by authorised parties in cases where the corresponding decryption keys are not otherwise available [8]. This usually involves a Trusted Third Party (TTP) that has the capability of restoring the appropriate decryption keys and this process is generically called *key recovery* (KR). Two typical scenarios where KR may be needed are:

- when the decryption key has been lost or the user is not present to provide the key.
- where commercial organisations want to monitor their encrypted traffic without alerting the communicating parties, for example to check that employees are not violating the organisation's policies.

National governments have also shown interest in the deployment of key recovery techniques, mainly motivated by law enforcement and intelligence concerns about the reduction in their capability for wiretapping when strong cryptography is used. This concern has led to proposals such as the famous Clipper Chip in the USA and the GCHQ protocol in the UK [USD94, CES96].

In this paper we use the following terminology:

Key Recovery Agent (KRA) Trusted third party that performs KR in response to an authorised request.

Key Recovery Information (KRI) Aggregate of data that is needed by the KRA in order to complete a KR request, e.g. a session key encrypted under the KRA's public key.

Key Recovery Requester (KRR) Authorised entity that requests KR from the KRA. The KRR would usually be a LEA in possession of a valid warrant.

Interception Agent Entity that acts in response to an authorised request for interception on a target identity by filtering out the communications traffic corresponding to such target identity. This function would usually be performed by NOs [13,11].

Rantos and Mitchell [17] proposed a KR scheme for UMTS as part of the European Community research project on Advanced Security for Personal Communications Technology (ASPeCT) [2]. The authors' strategy was to modify the already designed and well-studied ASPeCT WAKE protocol for the user-to-VASP interface [14]. These authors identified the following goals and requirements for the KR enhanced version of the WAKE protocol:

- R1 Recoverability of the session key K by the corresponding KRA. This is clearly the main goal: at the end of a successful run of the protocol the KRA that each entity is associated with should be capable of restoring K when presented with the available KRI.
- R2 Minimal computational overhead. An overriding constraint in mobile communications is the limited computational power of the mobile equipment since usually cryptographic operations will be performed on smart cards. Rantos and Mitchell also state that the computational overhead at the user end should be kept at the same level.
- R3 Unobtrusiveness. The enhanced protocol should not introduce any vulnerability into the ASPeCT protocol. This would occur if any of the security services provided originally by the protocol was compromised by the added KR mechanism. Obviously we have to make an exception to this requirement, with regard to the confidentiality of the communications which, in the KR enhanced version, can be revoked by the corresponding KRAs.
- R4 Fine granularity. The number of session keys that can be recovered from a single instance of KRI should be small enough so as to ensure fine granularity of authorised interception periods. In other words, KR that have been authorised for a specified period should not compromise communications outside the scope of the authorisation.

In the following two sections we describe and analyse the properties of the KR enhanced ASPeCT protocol. It turns out that at least one of the above requirements is actually not met by the protocol. In particular the unobtrusiveness requirement is not achieved, which allows an impersonation attack in the protocol. We also propose a fix to prevent such an attack as well as a new KR mechanism which exhibits additional and improved properties.

3 Key Recovery for ASPeCT Protocol

3.1 ASPeCT Protocol

The most well known public key based WAKE protocol for UMTS is proposed by the ASPeCT project, which is responsible for the research and development of security technologies to be used in the UMTS system. We shall call this protocol the ASPeCT protocol in this paper. The ASPeCT protocol is proposed for both user-to-network and user-to-VASP interfaces to enable economical deployment of security services. Detailed descriptions of the ASPeCT protocol can be found in the literature [14,1]. The message flows are shown in figure 2, where all the charging related data fields are omitted for simplicity. The notation used in this, and subsequent protocol descriptions, is shown in table 1.

A		the identity of the user
B		the identity of the VASP
TTP_A		the identity of the TTP of user A
g		a generator of a finite group
r_A		a random nonce chosen by user A
r_B		a random nonce chosen by the VASP, B
K_{AB}		a secret session key established between the user and the VASP, B
A_{Cert}		public key certificate of the user, A
B_{Cert}		public key certificate of the VASP, B
b		the private key component of the public-private key-agreement key pair of the VASP, B
g^b		the public key component of the public-private key-agreement key pair of the VASP, B
$\{m\}_{K_A^{-1}}$		the message m signed by the user with his/her private signature key K_A^{-1}
$\{m\}_{K_{AB}}$		the symmetric encryption of a message m using the session key K_{AB}
h_1, h_2, h_3		one-way hash functions

Table 1. Notation for Protocol Descriptions

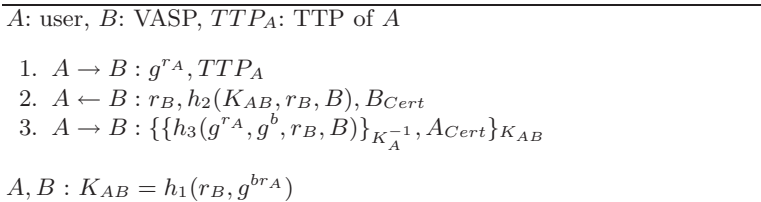


Fig. 2. The ASPeCT protocol

According to Horn and Preneel [14], the protocol satisfies the following six goals:

1. mutual explicit authentication of A and B ;
2. agreement between A and B on a secret session key K_{AB} with mutual implicit key authentication;
3. mutual key confirmation between A and B ;
4. mutual assurance of key freshness (mutual key control);
5. non-repudiation of origin by A for relevant data sent from A to B ;
6. confidentiality of relevant data sent by A to B .

The ASPeCT protocol is related to the Station-to-Station (STS) protocol of Diffie *et al.* [10] in that both protocols use the same challenge-response mechanism, i.e., A and B challenge each other with random nonces (g^{r_A} and r_B in this protocol) exchanged in clear and calculate responses using private keys (K_A^{-1} and b respectively in this protocol). In this protocol, however, it should be noted that the second message in the protocol does not contain any signature by B . The third message includes the signature by A like STS protocol to accommodate non-repudiation requirement for the relevant message from the user. In fact, the protocol shown in figure 2 is one of two variants of the ASPeCT protocol for user-VASP application, which is called variant B. Another variant, so called variant C, is an extended version of the protocol to include on-line TTPs, where the first message from A to B include the user identity encrypted under the key $L = (g^{x_A})^{r_A}$, where g^{x_A} is the public key agreement key of KRA_A . This new field allows the TTP of user A to identify the user, verify whether the user's certificate has been revoked, and deliver the user's certificate A_{Cert} to the VASP.

3.2 KR Enhanced ASPeCT Protocol

In the KR enhanced ASPeCT protocol proposed by Rantos and Mitchel [17], each entity A and B , registers with a KRA, KRA_A and KRA_B , in their respective domains. The same TTP is assumed to act both as the certification authority (CA) and the KRA for each entity. Two different solutions for KR are proposed that can be applied to both variants of the ASPeCT protocol. For brevity, we only describe the B-variant protocol. The extrapolation to the C-variant protocol is straightforward. Figure 3 illustrates the first of the two given solutions.

The KR capability is achieved by modifying the way in which r_A is generated. Specifically, r_A is now computed as

$$r_A = f(w_A, s_A)$$

where f is a one-way function, s_A is a one-time random seed, and w_A is a secret value shared between A and KRA_A which has been previously established between the two of them during the registration phase. Message 1 also includes A 's identity encrypted under the key $L = (g^{x_A})^{r_A}$, where g^{x_A} is the public key agreement key of KRA_A .

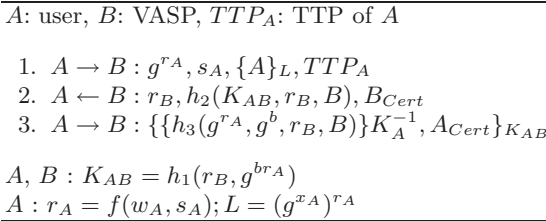


Fig. 3. KR Enhanced B-Variant Protocol

A request for key recovery to KRA_A will include the public values s_A , $\{A\}_L$, r_B and g^b . With this information, KRA_A can decrypt $\{A\}_L$ and, from A 's identity, obtain the corresponding secret key w_A , which in turn can be used to recompute K_{AB} . In B 's domain the situation is different. B escrows his private key b with his associated agent KRA_B during the registration phase. Thus KRA_B can restore K_{AB} when presented with r_B and g^{r_A} . The requests for KR will have to be accompanied by the appropriate authorisation (warrant). Clearly, other fields from the above protocol will also have to be submitted together with the request so that the KRA can check that the request for KR is within the scope of the warrant. However, for the sake of brevity we omit such details.

As a further enhancement Rantos and Mitchell [17] point out that w_A can be a temporary secret computed using a second one-way function f^* as

$$w_A = f^*(w_A^*, TT)$$

where w_A^* is a long-term secret shared with KRA_A , and TT is a date stamp. With this enhancement, KRA_A can delegate her capability to carry out key recovery to authorised parties, such as law enforcement agents, for the periods of validity of the values w_A . Thus, when an authorised request for the interception of A 's messages during a certain period is presented to KRA_A , instead of having to participate in the recovery of individual session keys, KRA_A can give the appropriate w_A values to the interception agents so that they can recover the keys themselves. This clearly makes more flexible and efficient the way in which wiretapping is performed. Unfortunately, the same does not apply to B 's domain, since b is the long-term secret key of B and consequently cannot be disclosed, for it would compromise both past and future communications.

A slight variation is also described [17] that includes s_A in the encryption of the first message as

1. $A \rightarrow B : g^{r_A}, \{A, s_A\}_L, TTP_A$

We notice though that in this case the enhancement described above where w_A is a fixed term secret cannot be used. The KRA would still need to cooperate in the recovery of all session keys, for only he can obtain s_A .

The second solution proposed by the authors does not require any shared secret. Again, only the first message is changed:

1. $A \rightarrow B : g^{r_A}, \{A, r_A\}_L, TTP_A$

Here r_A is again a randomly chosen value. The KRA can obtain r_A directly by decrypting the second field in the message, and therefore recompute K_{AB} .

4 Analysis of the KR Enhanced ASPeCT Protocol

In this section we study the properties of KR enhanced ASPeCT protocol. In particular we investigate whether the requirements defined in Section 2 are satisfied by such protocol.

R1. Session Key Recoverability The first of the requirements is clearly achieved. If the protocol is executed correctly, both KRAs can independently restore the session key established by A and B , by proceeding as described above.

R2. Computational Overhead Since modular exponentiations are the operations that consume the most computational resources, we use the number of exponentiations as an indicative measure of the computational complexity of the protocols. Thus, we observe that for the KR enhanced B-variant protocol the introduced computational overhead consists of an extra exponentiation on A 's side in computing $L = (g^{x_A})^{r_A}$, which in turn is used to encrypt A 's identity. Since the C-variant protocol already calculated L , no extra-exponentiation is introduced in the KR enhanced version of the protocol.

The authors reason that explicitly adding A 's identity in the KR enhanced B-variant protocol allows KRA_A to obtain the corresponding secret key w_A . The identity is further encrypted under L in order to preserve the anonymity of the user. Notice that in the original protocol the user remains unidentified to B until he receives the third message. However, it is our contention that such a field is altogether unnecessary. To see this we have to realise that anonymity revocation has to occur prior to KR, when the encrypted communications are intercepted. In other words, the interception agents that want to wire-tap some specified target user's communications need to be able to discriminate such communications from the rest of communications that occur simultaneously. Once this is done, the intercepted KRI can be used to request KR from the KRA. Notice that since A 's identity is encrypted under L , which is only known to A and KRA_A , the interception agents will not be able to use that field to discriminate communications based on a target identity and, therefore, a different mechanism outside the scope of the protocol would be required.

It seems that a likely way in which the filtering would be performed is by requiring the cooperation of the NOs [11]. Recall that during the set-up phase of a communications association between A and B , both users authenticate themselves to their respective NOs. Hence NOs are the obvious candidates for filtering encrypted communications based on target identities. They could hand over the intercepted encrypted data to the authorised LEAs or, even more, interact as a proxy between the LEAs and the KRAs. In any case since the identification of the encrypted data has been performed before the actual submission of the KRI

to the KRA, the inclusion of $\{A\}_L$ in message 1 seems unnecessary, for the identity of the target user is passed by the LEA to the KRA as part of the warrant. We note that this must have also been the assumption of Rantos and Mitchell [17] when they pointed out the possibility of delegating the KR capability to authorised requesters by giving them w_A , in the case where w_A is a temporary value.

Hence, from the above argument, we see that the field $\{A\}_L$ can be safely dropped from message 1. This will save an expensive exponentiation if the first KR mechanism is used in its first variant, i.e. with s_A being sent in the clear. Thus the first message of the protocol becomes:

1. $A \rightarrow B : g^{r_A}, s_A, TTP_A$

Obviously when the second KR mechanism is used with the B-variant protocol L has to be computed, for r_A still must be encrypted using it.

R3. Unobtrusiveness A significant defect in the above protocol is the failure to satisfy requirement R3 for unobtrusiveness. Strictly speaking, the mutual authentication service that was originally provided is sacrificed when the KR capability is added to the WAKE protocol. User A cannot be sure whether the protocol messages are being exchanged with B or KRA_B , for both know b . Even worse, in the case where w_A is a temporary secret an impersonation attack can be mounted as explained below.

Attack An attacker C with knowledge of w_A can impersonate B to A during an authorised interception period. After intercepting message 1, C calculates $r_A = f(w_A, s_A)$, chooses a random value r'_B , and computes the session key as

$$K'_{AB} = h_1(r'_B, (g^b)^{r_A})$$

He proceeds by forming message 2 as shown in Figure 4 and sending it to A , effectively impersonating B .

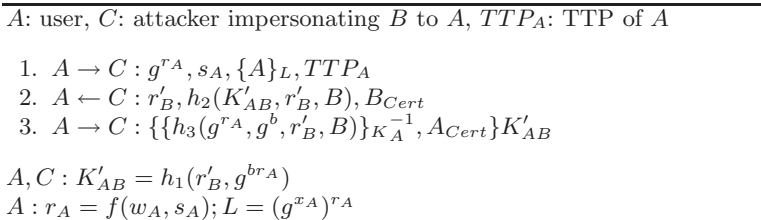


Fig. 4. Impersonation attack in the KR Enhanced B-Variant Protocol

The above attack can be easily fixed by simply holding s_A until message 3 of the protocol. Even with this modification, R3 is not satisfied. In order to

comply with requirement R3, escrow of b has to be avoided. With this condition, we notice that the secret information that is needed by B 's KRA in order to be able to recover the session key is g^{br^A} . Hence B needs to somehow make possible KRA_B 's access to such piece of data. At first thought, we may suggest to convey it in message 2 of the protocol. For example, B generates r_B in the same way as A , i.e. $r_B = f_B(w_B, s_B)$, where w_B is a secret value shared with KRA_B , and s_B is a random number; and then sends $r_B \oplus g^{br^A}$ and s_B in message 2. However the same kind of impersonation attack can be mounted yet again. When an interception agent with temporary knowledge of w_B sees messages 2, he intercepts it and extracts g^{br^A} by first calculating

$$r_B = f_B(w_B, s_B)$$

and therefore

$$g^{br^A} = r_B \oplus (r_B \oplus g^{br^A}).$$

Now he can generate a new random number r'_B , and impersonate B by generating a new bogus message and sending it to A . Hence we see that B cannot give his KRA access to g^{br^A} in message two. But message two is the only one that he produces in the protocol. An obvious solution entails sending a fourth message with the involved communications overhead. Alternatively, we can require the cooperation of A . For example, instead of sending s_B in the clear, B could send $\{s_B\}_{K_{AB}}$ in message two. On receiving it, A would decrypt s_B and include it in message 3. Figure 5 shows a modified version of the KR enhanced B-variant protocol with all the suggested alterations. Notice that s_A is withheld until message 3 to counteract the first impersonation attack described above. Also, we have dropped the encryption of A 's identity under L in message 1 in accordance with our discussion on requirement R2.

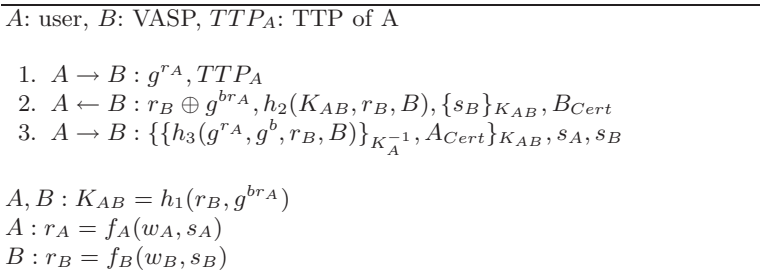


Fig. 5. Modified KR Enhanced B-Variant Protocol

When A receives the second message she extracts r_B , using the value $(g^b)^{r^A}$. With r_B she can compute K_{AB} and obtain s_B by decrypting the second last field of message 2. The authentication of B to A is finalised when A checks that the hash value of message 2 corresponds to the value that she has calculated for K_{AB} .

R4. Fine Granularity We can easily verify that the requirement for fine granularity is achieved by the KR enhanced protocols. In the case where w_A is a long term secret, the keys are recovered by the KRAs in an individual basis in both domains, and the recovery of any subset of them does not compromise any other session key. In the case where w_A is a temporary secret in A 's domain, the period of validity of w_A can be adjusted to yield any convenient granularity.

5 Enforceability

Enforceability refers to the safeguards that a KR scheme might provide in order to ensure that the KR mechanism cannot be circumvented by the users of the scheme. This could happen if users succeed in communicating confidentially without allowing the recovery of the relevant session key. The actual scope of any enforceability mechanism depends on the threat model used for the KR system. Thus, we can distinguish two different enforceability levels:

Level 1 At this level, the enforceability mechanism ensures that no user of the system (including VASPs) can succeed in circumventing the KR mechanism unilaterally, i.e. without the cooperation of the other communicating party.

Level 2 At this level, the enforceability mechanism ensures that no user of the system (including VASPs) can succeed in circumventing the KR mechanism, even with the cooperation of the other communicating party.

In the ASPeCT protocol, if user A does not follow the protocol accurately and sends a bogus value as the second field in the first message, then the session key will not be recoverable in A 's domain. This stems from the lack of enforceability of the KR mechanism in A 's domain. The same cannot be said of B 's domain, at least at a level 1 enforceability. Here, KR is enforced by the certification process itself. B is refused the certification of his public key agreement key if the corresponding private key is not escrowed.

When the KR mechanisms are used with the C-variant protocol, Rantos and Mitchell [17] point out that, if required, B can relay the first message to A 's on-line TTP who then validates that the correct KR field was sent by reconstructing g^{r_A} . This helps avoid single rogue user attacks. Thus the TTP who, let's recall, is both A 's CA and KRA, can additionally perform a KRI validation function. More specifically we define such a type of function as a *private KRI validation function* since in order to validate the KRI it is necessary to have knowledge of access-restricted information, in this case w_A .

Similarly we can define a *public KRI validation function* as a validation function that can be executed by anyone using only publicly available information. The provision of one such function would have the advantage of not requiring the involvement of the KRA. This makes that function easily distributable, shifting away the heavy burden placed on the KRAs if they were to oversee all the communications of their associated users. In the next section of this paper we present a refined KR mechanism that provides a public KRI validation function without requiring an increase in the computational load of A and B . Thus, with

the new KR mechanism, validation can be performed efficiently in both the B and C-variants of the ASPeCT protocol. Furthermore, there would be no need for the TTP to act both as the CA and the KRA, as far as KR validation is concerned.

Whether enforceability, at any degree, is a requirement for KR in the wireless communications arena is an open question. A pervasive problem with KR, which applies to all techniques, is that of super-encryption [5]. From a cryptographic point of view, once an authenticated communications channel is provided, establishment of non-recoverable keys is a trivial issue. For example, an authenticated channel is all that is required to securely run the Diffie-Hellman key agreement protocol [9], which could be executed at the application layer to establish a “parallel” session key that is not recoverable by the KRAs and which is used to encrypt the data (also at the application layer) before passing it to the communications layer where the protocols described above operate. Thus, even if the session key established at the communications layer is restored by a KRA, access to the plaintext data is still not possible. In other words, Level 2 enforceability does not appear achievable, at least without the utilisation of trusted functionality (e.g. tamperproof implementations) to thwart misuse of the system. On the other hand, even if misuse of the system is technically possible, the wiretapping capability left in practice to law enforcement agencies may still make the deployment of such KR system worthy.

6 A New KR Mechanism

As already mentioned, a more flexible option to the validation mechanism proposed in [17] would be to provide a public KRI validation function. This allows any third party to detect misuse of the protocol by using only public data. For example, in the wireless environment, NOs could be given the task to perform the validation function. Thus, if required, they could enforce the KR mechanism by aborting any communications in cases where the KRI cannot be validated successfully. It is important to realise that still, the only entities that could recover session keys would be the KRAs. In this section we describe a new KR mechanism that provides a public KRI validation function and that could be easily incorporated to the ASPeCT protocol in A 's domain.

Firstly we note that the new KR mechanism is applicable in cases where the target key to be recovered is of the form

$$K_{AB} = f(r, \text{other public information})$$

where f is a publicly known one-way function and r is a secret random number generated by the user. In these cases, key recovery is equivalent to restoring r .

In the description of the KR mechanism we use the following notation: p a large prime, q a prime with $q|p-1$, and g an element in the multiplicative group \mathbb{Z}_p^* of order q . All operations are performed modulo p , except where noted otherwise. The KR mechanism consists of three stages as follows.

KRI Generation phase The user, say A does the following.

1. Generate a secret KR key w , $1 \leq w \leq q - 1$, which she shares with her associated KRA. Optionally w can be generated by the KRA, or agreed by, both parties. The value $\phi = g^w$ is made publicly available. This step can be a one-off process, or alternatively, repeated at any convenient frequency.
2. Select a random integer r , such that $1 \leq r \leq q - 1$, and compute $u = g^r$.
3. Compute $c = h(u) \bmod q$, where h is an appropriate hash function.
4. Compute $s = wc + r \bmod q$.

The KRI in A 's domain is $\{w, u, s, A\}$ of which $\{u, s, A\}$ are public.

Public KRI Validation phase Given the public input data u, s, A , a monitoring third party V can check the integrity of the KRI fields generated by a user A , by doing the following:

1. Obtain authentic public value ϕ .
2. Compute $c' = h(u) \bmod q$.
3. V resolves the validation process as successful if and only if $g^s = \phi^{c'} u$.

KR phase Provided that the KRI verification function is successful, the corresponding key recovery agent can recover the value r_i when presented with the input data $\{s, u, A\}$ by doing the following:

1. Obtain w corresponding to user A .
2. Compute $c = h(u) \bmod q$.
3. Compute $r = s - wc \bmod q$.

6.1 Security of the KR Mechanism

The security of the above mechanism can be reduced to that of the non-interactive proof of possession of $\log_g \phi_i$ as implemented in Schnorr's signature scheme [18] whose properties are well known. When user i generates $\{s_i, u_i\}$, she produces a proof of knowledge of w . Its security relies on the following two propositions:

1. Knowledge of w implies the capability of recovering r , and vice-versa.
2. Provided that h can be assumed to yield random values, knowledge of $\{\phi, u, s\}$ does not give any knowledge about w , no matter how many times we reuse ϕ to produce such triplets.

It is interesting to note that we may regard r as a *verifiable encryption* of the discrete log of the public value s . The verification is much more efficient than other more general verifiable encryption protocols [19]. The reason that this is possible is that in our case the verifier never actually obtains the value r (which would give away the shared secret w). This means that our mechanism is not applicable in applications such as fair exchange [3].

6.2 ASPeCT Protocol with New KR Mechanism

It is easy to see that the above mechanism can be used in the ASPeCT protocol by making $\phi_A = g^{w_A}$ public and changing the way r_A and s_A are calculated. Now, according to the above mechanism r_A is a random value and s_A is calculated as

$$s_A = w_A h(g^{r_A}) + r_A \bmod q$$

The rest of the protocol is the same. Now any monitoring third party can check the validity of the KRI formed by A using the KRI validation function described above. Hence, there is no need to require A 's TTP to be on-line in order to validate the KRI, which makes the above mechanism suitable for both the B and C-variants of the ASPeCT protocol.

Unfortunately, due to the different way in which B authenticates to A , the same KR mechanism cannot be applied in B 's domain in the modified KR enhanced ASPeCT protocol that we proposed in Section 4. This would not happen for other protocols which are more symmetric, such as the STS protocol [10].

7 Multiple KRAs

When designing KR schemes, it is common practice to distribute the trust vested in the KRA functionality among multiple KRAs, i.e. users have several associated KRAs that have to cooperate in order to perform KR. This helps increasing the users' acceptability on the KR system. We observe however that, contrary to most KR proposals, the ASPeCT KR scheme specifies only one KRA. Therefore, it is possible for a single KRA to wiretap on users communications. Notice that a KRA can revoke the identity of any user she is associated with in polynomial time by simply testing all the possible identities.

A simple example of how to allow multiple KRAs using the modified KR enhanced ASPeCT protocol (Figure 5) is as follows.

A user, say A , establishes a secret value w_i with each KRA, KRA_i ($i = 1, \dots, n$). For each run of the protocol, A calculates

$$r_A = f_A(w_1, s_A) \oplus f_A(w_2, s_A) \oplus \dots \oplus f_A(w_n, s_A)$$

The rest of the protocol remains the same. An authorised requester seeking KR does the following:

1. Contacts each KRI_i and presents the appropriate authorisation information, together with the values s_A and A corresponding to the intercepted communication. KRI_i then calculates $v_i = f_A(w_i, s_A)$, which she returns to the requester.
2. Once all the KRAs have been contacted, she restores r_A as:

$$r_A = f_A v_1 \oplus v_2 \oplus \dots \oplus v_n$$

3. Finally, since r_B, g^b are public information, she can compute the session key as:

$$K_{AB} = h_1(r_B, g^{b^{r_A}}).$$

8 Conclusion

In this paper, we analysed a key recovery proposal for the ASPeCT protocol and identified several weaknesses. A modification was proposed that fixes the weaknesses and, exhibits additional and improved properties.

References

1. ACTS AC095, ASPeCT Deliverable D02, Initial Report on Security Requirements, AC095/ATEA/W21/DS/P/02/B, Feb., 1997, Available from <http://www.esat.kuleuven.ac.be/cosic/aspect/>. 227
2. Advanced Security for Personal Communications Technologies. <http://www.esat.kuleuven.ac.be/cosic/aspect/index.html> 226
3. N.Asokan, V. Shoup and M. Waidner, Optimistic Fair Exchange of Digital Signatures, Eurocrypt'98, Springer-Verlag, 1998, pp.591-606. 235
4. U. Black, Third Generation Mobile Systems (TGMSs), in Second Generation Mobile & Wireless Networks, Parentice Hall, 1999. 223
5. B. Pfitzmann and M. Waidner, How to Break Fraud-Detectable Key Recovery, Operating Systems Review, 21, 1998, pp.23-28. 234
6. K. Buhanal et al., IMT-2000: Service Providers Perspective, IEEE Personal Communications, August 1997. 223
7. CESG, Securing Electronic Mail within HMG - part 1: Infrastructure and Protocol, document T/3113TL/2776/11, March 1996, available at <http://www.rdg.opengroup.org/public/tech/security/pki/casm/casm.htm>.
8. Denning D. and Branstad D., A Taxonomy for Key Escrow Encryption systems, Communications of the ACM, Vol. 39, Pp 34-40, 1996.
9. Diffie W. and Hellman M., New Directions in Cryptography, IEEE Transactions on Information Theory, 22, pp 644-654, 1976. 225 234
10. W.Diffie, P. van Oorschot and M.Wiener, Authentication and Authenticated Key Exchanges, Designs Codes and Cryptography, 2, pp.107-125, 1992. 228, 236
11. ETSI TC-STAG, "Security Techniques Advisory Group (STAG); Definition of User Requirements for Lawful Interception of telecommunications; Requirements of the Law Enforcement Agencies", ETR 331, December 1996. 226, 230
12. ETSI SMG10, Draft UMTS 33.21 version 2.0.0, Universal Mobile Telecommunications System (UMTS): Security Requirements, Feb., 1999. 225
13. ETSI TC Security, Specification for Trusted Third Party Services: Part1 Key Management and Key Escrow/Recovery, DEN/SEC-003001x, Draft Version 1,0 (edition2), 11th, Nov. 1997 226
14. Horn G. and Preneel B., Authentication and payment in future mobile systems, Computer Security - ESORICS'98, Lecture Notes in Computer Science, 1485, pp. 277-293, 1998. 226, 227, 228
15. K. Martin, Applying Cryptography within the ASPeCT Project, Information Security Technical Report, 1998, Vol. 2, No. 4, pp. 41-53. 223
16. T. Ojanpera and R. Prasad, IMT-2000 Applications, in Widenband CDMA for Third Generation Mobile Communication, T. Ojanpera and R. Prasad (ed.), Artech House Publishers, 1998, pp. 65-76. 223
17. Rantos K.and Mitchell C., Key recovery in ASPeCT authentication and initialisation of payment protocol, Proceedings of ACTS Mobile Summit, Sorrento, Italy, June 1999. 226, 228, 229, 231, 233, 234
18. Schnorr C.P., Efficient Identification and Signatures for Smart Cards- CRYPTO'89, Proceedings, Lecture Notes in Computer Science, vol 330, Springer-Verlag, pp 239-251, 1990. 235
19. M. Stadler, Publicly Verifiable Secret Sharing, Eurocrypt'96, Springer-Verlag, 1996, pp.190-199. 235
20. US Department of Commerce, National Institute of Standard and Technology, FIPS PUB 185, Escrowed Encryption Standard, February 1994.