Key Recovery in ASPeCT Authentication and Initialisation of Payment Protocol

Konstantinos Rantos* and Chris Mitchell
Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK.
K.Rantos@dcs.rhbnc.ac.uk, C.Mitchell@rhbnc.ac.uk

28th January 1999

Abstract

This paper seeks to give solutions to possible demands for lawful interception of communications. Certain modifications to the ASPeCT Authentication and Initialisation of Payment protocol are proposed that give it a key recovery capability. The modified protocol fulfils potential government requirements for lawful interception while protecting the user from unauthorized disclosure of his/her communications.

Keywords: UMTS, key recovery.

1 Introduction

The growth of telecommunications has created a clear demand for lawful interception, mainly for the investigation of serious crime and for national security reasons. Before the employment of encryption for the protection of communications, access to transmitted data was just a matter of wire-tapping or listening to the air interface. The introduction of confidentiality services for protecting communications and archived data has created the need for key recovery (escrow) services [1].

This paper proposes certain modifications to the ASPeCT (Advanced Security for Personal Communications Technology) Authentication and Initialisation of Payment (AIP) protocol that give it a key recovery capability. The modified mechanism gives Law Enforcement Agencies (LEAs) access to transient keys and therefore offers the capability of accessing, when authorized, suspected communications while protecting the user from unauthorized disclosure of his/her data. LEAs will only be able to access the communications they are authorized to.

*This author's work is supported by the European Commission (TMR Marie Curie Research and Training Grant ERBFMBICT983274).

2 The ASPeCT AIP Protocol

Among the authentication schemes proposed for third generation mobile systems is the one designed and implemented by the collaborative research project ASPeCT. The ASPeCT AIP protocol was developed for authentication between a user U and a value added service provider (VASP) V in Universal Mobile Telecommunications System (UMTS) environments. Two basic models have been designed for this purpose (B and C variants).

2.1 Authentication without an on-line TTP (B-Variant)

A detailed description of this model is given in [3] and the messages exchanged are specified in Fig.1.

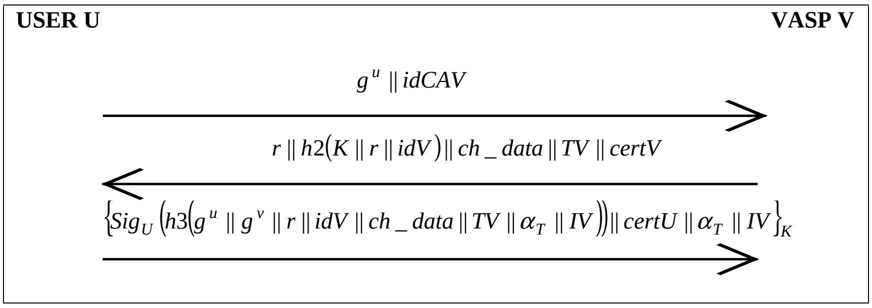

Figure 1: ASPeCT AIP Protocol (B-Variant)

In this model U generates a random number u, computes g^u and sends it to V together with the identity $idCAV$ of the authority whose certificates U can verify. On receipt of the first message V generates a random number r and computes a session key $K = h1((g^u)^v \| r)$ where v is V's private key agreement key and $h1$ a hash function. V then sends U the random number r, the hash value $h2(K \| r \| idV)$ and its certificate $certV$ together with a time-stamp TV and charging-relevant data ch_data. On receipt of the second message, U computes the key $K = h1((g^v)^u \| r)$ and compares the hashed value $h2(K \| r \| idV)$ with the one received. If the check succeeds U generates the signature shown in Fig.1, including random number IV and $\alpha_T = F_{IV}^T(\alpha_0)$, where α_0 is random, as required by the payment protocol, and sends the last message encrypted with K.

2.2 Authentication with an on-line TTP (C-variant)

The second authentication model involves an on-line TTP. The protocol described is an adaptation of the one published in [4] and has the same properties as the ones in [6] and [2]. The messages exchanged are specified in Fig.2 and a full description and analysis of the protocol is given in [3].

In this variant of the protocol U sends V the value g^u together with the identity $idTTP$ of his TTP and his own identity idU encrypted under session

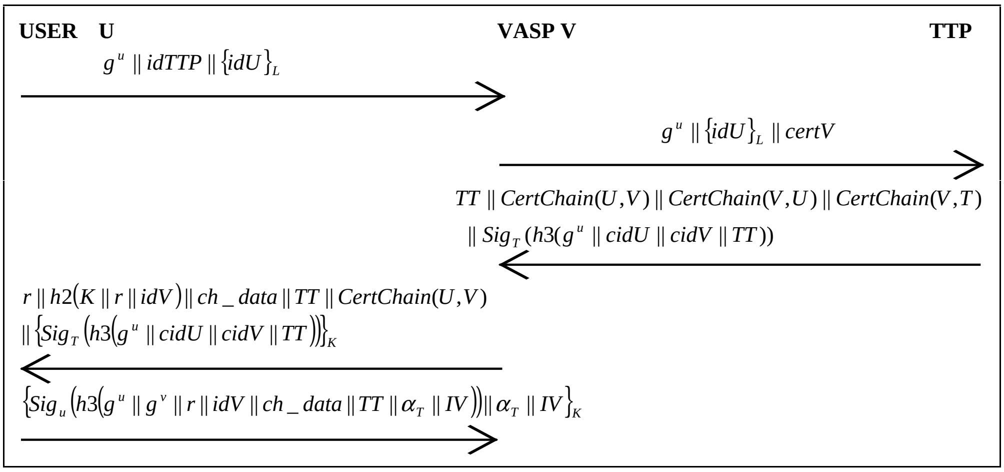

Figure 2: ASPeCT AIP Protocol (C-Variant)

key $L = g^{uw}$, where g^w is TTP's public key agreement key. As soon as V receives the first message it connects to U's TTP and forwards the message sent by U together with its certificate $CertV$. On receipt of the second authentication message the TTP checks whether U's and optionally V's certificates have been revoked. If both certificates are valid, the TTP generates the certificate chains and sends them back to V together with a time-stamp TT and a signature on the certificate identifiers $cidU$ and $cidV$, the time-stamp TT and the random number g^u. V verifies $CertChain(V, U)$ and the signature using the TTP's public key which retrieves from $CertChain(V, T)$. It computes a hash value on the session key K concatenated with the random number r and V's identity idV. V also encrypts the signature with key K. V then forwards to U the encrypted signature together with the hash value $h2(K \parallel r \parallel idV)$, the cross-certificate for V's public key $CertChain(U, V)$, the random number r, the time-stamp TT and charge data ch_data. On receipt of the fourth authentication message U decrypts the signature, checks its validity and that of the cross-certificate, and if the checks are successful U responds with the fifth authentication message.

3 Requirements and Goals for Key Recovery in the ASPeCT Protocol

Among the properties of the ASPeCT AIP protocol is the establishment of a secret session key $K = h1(g^{uv}, r)$. The enhanced protocol should give the TTP, which acts as a Key Recovery Agent (KRA), the ability to recover the requested session key K when provided with the appropriate key recovery material. One of the main requirements of the key recovery mechanism employed is to keep the computational overhead at the user end at the same level. This is desirable because all the user computations are typically performed by a smart card. An effective solution would therefore be to make the key recovery mechanism part of the key establishment process without introducing any vulnerabilities. In this paper two different solutions to the key recovery problem are proposed. Although both solutions apply to both basic models of the ASPeCT protocol, for brevity we apply one solution to each model.

3.1 B-variant protocol with key recovery capability

The B-variant can be given a key recovery capability by slightly modifying the way that U's key component u is generated. Note that, in the existing variants of the protocol, the value u is chosen at random by U prior to the start of the protocol.

The user's key component generation becomes a two-phase procedure. First, there is a *key recovery registration* phase where the user registers with his TTP, in an escrow-like mechanism, an initial secret key value k_u. Second, each time the user wants to generate a key component, the *key generation* phase, he/she generates a random (or serial) number s and combines s and k_u to get the key component u. That is, $u = f(k_u, s)$ where f should be a one way function (cf. the requirements given in clause 6 of ISO/IEC 11770-3 [5]). In order for the TTP to be able to compute the value u, U has to send the TTP his own identity idU and the value s encrypted under $L = (g^w)^u$, where g^w is the TTP's public key agreement key. The modified scheme therefore, requires the TTP to have a key agreement key, as in the C-variant. Thus, the modified protocol is as specified in Fig.3.

Figure 3: Modified B-variant Protocol

In U's domain, the keys can be recovered as follows.

- The entity requesting key recovery gives U's TTP, which acts as a KRA, the following intercepted values:

  1. The one-time random value g^u, V's certificate $certV$, the random number r and the encrypted value $\{idU \| s\}_L$. The TTP, using the value g^u and its private key agreement key w can compute the session key L and therefore decrypt the value $\{idU \| s\}_L$. This will enable the TTP to compute the value u and, having already the values r and g^v, to recover the key K and send it to the requesting entity.
  2. The last authentication message sent by U to V together with the charging data *ch_data* and the time-stamp TT. These values will help the TTP verify U's signature so that it can check that the request is within the scope of the warrant.

More generally, a second one-way function f^* could be employed to increase flexibility. The user would keep a long term secret k_u^* (also known to the user's TTP). From this value the user would compute a 'fixed term' secret k_u, by combining k_u^* and a date stamp using f^*. In such a case the TTP could disclose the value k_u for a particular time period to the intercepting authority, and would thereby only reveal the user's key values u for a fixed time interval.

In V's domain, however, the procedure is slightly different, and the key recovery process is less flexible. This is because it would typically not be desirable to send the user's secret key component to V's TTP (especially when U's and V's TTPs are in different domains or simply when V's TTP is not trusted by the user). Therefore, V has to register with its TTP the private key agreement key v. This can be done at the time a certificate on the public key agreement key g^v is requested and issued. Thus, in V's domain the key recovery procedure is almost the same as in U's domain. The only difference is the way that V's TTP recovers the session key K. However, the flexibility provided in the user's domain is no longer available, since if V's private key agreement key v is revealed, then all previous and subsequent communications to and from the VASP can be decrypted. In most scenarios this will be inappropriate, so the TTP must pass to the entity requesting recovery only the session key K.

Finally note that the value s could also be sent in clear (and not encrypted under L). In such a case the function f must have the property that, given the input value s, an adversary cannot get any information on the output u (without knowledge of k_u).

3.2 C-variant protocol with key recovery capability

In this section another solution to the key recovery problem is proposed which, as mentioned earlier, can also apply to the B-variant. Essentially, this variant gives a key recovery capability simply by passing the TTP the key component u encrypted under the secret key L. This gives the TTP the ability to recover the key K. Thus, the two first messages of the enhanced protocol (this is the only modification required) are as shown in Fig.4:

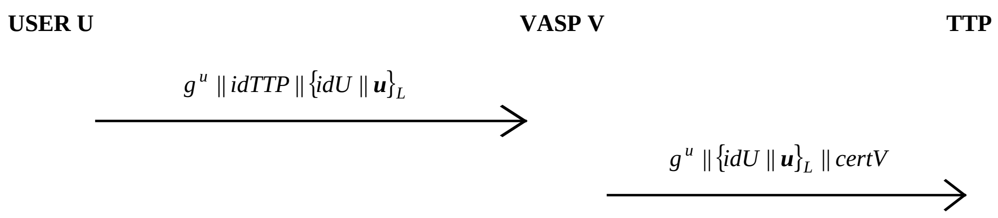

Figure 4: Modified Protocol Variant C6

In this solution, as mentioned earlier, U simply passes to its TTP the generated key component u encrypted under L. Thus, when intercepting the communication between the user and the VASP, all the information needed by the user's TTP to compute the session key K is available. The key recovery procedure is the same as in the previous solution both in U's and V's domain except for the session key K computation and the fact that the TTP's signature is sufficient to check that the request is within the scope of the warrant.

4 Conclusions

In this paper two mechanisms that give the ASPeCT AIP protocol a key recovery capability were proposed. The main requirements were to keep the changes required to a minimum and at the same time minimise the computational overhead at the user's end. The proposed mechanisms solve demands for warranted access to communications while protecting the user from further unauthorized disclosure of his/her data.

References

[1] Dorothy E. Denning and Dennis K. Branstad. A taxonomy of key escrow encryption systems. *Communications of the ACM*, **39**:34–40, March 1996.

[2] G. Horn, P. Howard, K.M. Martin, C.J. Mitchell, B. Preneel, and K. Rantos. Trialling secure billing with trusted third party support for UMTS applications. In *Proceedings of 3rd ACTS Mobile Communications Summit*, pages 574–579, 1998.

[3] G. Horn and B. Preneel. Authentication and payment in future mobile systems. In *Lecture Notes in Computer Science*, volume 1485, pages 539–548. Computer Security - ESORICS 98, 1998.

[4] G. Horn and B. Preneel. Authentication in future mobile systems. Technical Report KUL-ESAT-COSIC98-2, KUL-ESAT-COSIC, 1998.

[5] International Organization for Standardization, Genève, Switzerland. *ISO/IEC 11770-3, Information technology—Security techniques—Key management—Part 3: Mechanisms using asymmetric techniques*, 1998 (to be published).

[6] K.M. Martin, B. Preneel, C.J. Mitchell, H.J. Hitz, G. Horn, A. Poliakova, and P. Howard. Secure billing for mobile information services in UMTS. In *Lecture Notes in Computer Science*, volume 1430, pages 535–548. 5th International Conference in Services and Networks, IS&N'98, 1998.