20

Efficient and provably secure key agreement

N. Alexandris[1], M. Burmester[2], V. Chrissikopoulos[1] and D. Peppes[1]

[1] *Department of Informatics, University of Piraeus,*
80 Karaoli & Dimitriou Str. 185 34 Piraeus, Greece
tel. +30 1 4120751, fax +30 1 4179064
e-mail alexandr@unipi.gr, chris@unipi.gr and pepes@unipi.gr

[2] *Department of Mathematics, RH - University of London,*
Egham, Surrey, TW-20 OEX, UK
tel. +44 1784 443084, fax +44 1784 4430766
e-mail m.burmester@vms.rhbnc.ac.uk

Abstract

Data security in computer networks is becoming increasingly important due to the expanding role of distributed computation, distributed databases and tele-communication applications such as electronic mail and electronic funds transfer. For privacy, information which may be highly sensitive or privileged must be encrypted with secret keys which are shared by the communicating parties. These keys are generated by key agreement protocols. Traditionally such protocols were designed by trial and error. History has proven this method to be unreliable: many protocols were broken or serious flaws were exposed. In this paper we discuss the security aspects of key agreement protocols. In particular, we consider two models for provable security, one based on probabilistic encryption, the other on zero-knowledge. We propose a variant of the Diffie-Hellman key agreement protocol which is provably secure and efficient.

Keywords

Data security, key distribution, key agreement, provable security, zero-knowledge.

S. K. Katsikas et al. (eds.), *Information Systems Security*

1 INTRODUCTION

Security in computer networks is becoming increasingly important due to the expanding role of distributed computation, distributed databases and tele-communication applications such as electronic mail and electronic funds transfer. Data transferred may be highly sensitive or privileged information. Unauthorized access to this information is easy and cheap to obtain and difficult to detect if no security measures are taken.

When designing a network, several aspects of data security need to be considered. Prominent among these are spurious message injection, message reception by un- authorized receivers, transmission disruption and rerouting data to fake nodes. The services that a security system of a network has to provide are identification, peer entity authentication, access control, data confidentiality, communications integrity, service availability, accountability and non-repudiation (Needham-Schroeder 1978, Yao 1982, Diffie et al. 1992, Bellare-Rogaway 1994).

For this purpose a combination of authentication protocols and encryption algorithms on the data is utilized. Each encryption algorithm uses a secret key to encrypt the sensitive data. These keys have to be shared between the communicated entities. The problem of distributing securely keys is one of the main goals of network security. Two approaches may be used. In the first one, there is a Trusted Centre from which the entities obtain the required shared (session) keys. With such systems the Centre manages the system and has overall control, e.g. as with Kerberos (Kohl 1991). This is a major drawback, as too much trust is confined to one entity. The second approach is based on the use of public keys and appropriate protocols for generating shared (session) keys. Such systems have some intrinsic advantages and the role of the Centre is reduced to managing the public keys. In this paper we discuss the security aspects of key agreement protocols. In particular, we consider two models for provable security, one based on probabilistic encryption, the other on zero-knowledge. We propose a variant of the Diffie-Hellman key agreement protocol (Diffie-Hellman 1976) which is provably secure and has low complexity.

2 KEY AGREEMENT

Key Agreement protocols are used by the entities of a network to establish privately shared keys. After the execution of the protocol only authenticated entities can be in possession of the correct key or of the necessary information to compute the correct key (Diffie-Hellman 1976, Bellare-Rogaway 1994, ISO/IEC Draft Directory 1995).

The Key Agreement takes place in the structured setting of the network. Depending on the protocol used, the setting may be changed. Key Agreement protocols can be distinguished according to the required number of passes and the interactions. Traditionally, protocols were designed by trial and error. History has shown that the success rate of this method is not too impressive (Desmedt-Burmester 1993, Burmester 1994).

In the early 1980's many authors (Yao 1982, Goldwasser-Micali 1984, Blum-Micali 1984) suggested that the security of cryptographic schemes can be proven under standard and well believed complexity theoretic assumptions (e.g. the intractability of factoring). This entails providing (i) a definition of the goal, (ii) a protocol and, (iii) a proof that the protocol meets its goal, assuming some standard complexity-theoretic assumption holds true.

By the late 1980's provable security was achieved for probabilistic encryption, pseudo-random number generators and digital signatures (Yao 1982, Goldwasser-Micali 1984, Blum-Micali 1984, Goldwasser et al. 1988). The goal of building secure distributed systems is more complex. Since 1991, Bird et al. 1992, and Bellare-Rogaway 1994, addressed this issue and proposed protocols which offer varying degrees of protection.

The most popular Key Agreement schemes are based on a model proposed by Needham and Schroeder 1978. In these, the Trusted Centre selects a session key, encrypts it and sends it to the entities who wish to communicate in private. Leighton and Micali 1994 proposed a variant of this based on probabilistic encryption. The main drawbacks of these schemes are that the Trusted Centre knows all the secret keys and the session keys, and that it has to be online. Furthermore, arbitrary many cleartext-ciphertext pairs are exposed.

The Trusted Centre need not be continuously available if the secret keys are distributed in advance to all pairs of entities. This is known as the N^2 problem, where N is the maximum number of entities who want to communicate securely. Gong and Wheeler 1990 proposed a solution to this problem which reduces the overall number of the secret keys and the number of keys maintained by the entities. Leighton and Micali 1994 proposed a slightly different version.

In the following sections we will consider two models for provably secure key agreement. In both models we shall assume that the adversary as well as all the entities in the network have limited resources, bounded by a polynomial in the length of the shared session key.

3 PROVABLE SECURITY WITH PROBABILISTIC ENCRYPTIONS

A basic security requirement for key agreement is that a compromised session key should only affect the session which that key protects. Consequently the entities have to use a fresh session key each time that they want to communicate in private. An additional reason for freshness is the necessity of avoiding cross-session "replay attacks".

Bellare and Rogaway 1994 presented several key agreement protocols which are provably secure. These are based on probabilistic encryption and use pseudo-random functions which make it impossible for an adversary to distinguish their values from random strings. Furthermore, the adversary cannot combine such values to construct new ones, preventing "chosen-ciphertext" type attacks. In one of their proposed protocols, the entities agree on an implicit common session key. Figure 1 shows the messages exchanged between entities A and $\hat{A}$ in this protocol.

All pairs of entities are given privately by a Trusted Centre long-lived common secret keys which are $2k$-bit strings $(\acute{a}_1,\acute{a}_2)$. The first part $\acute{a}_1$ is taken as the key of a pseudo-random function f_{a_1} and is used for authentication. The second part $\acute{a}_2$ is taken as a key for another pseudo-random function f_{a_2} and is used to compute the session key. The main feature of this scheme is that it uses pseudo-random functions. A drawback is that many cleartext-ciphertext pairs encrypted under the same key are exposed.

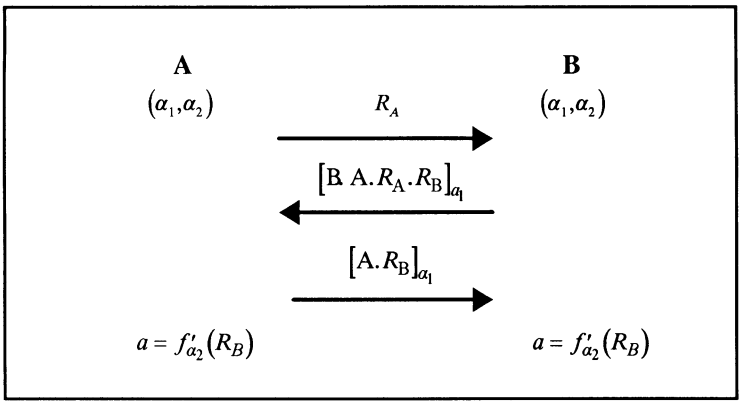

Figure 1 The Bellare-Rogaway Key Agreement protocol. Here $[x]_{\alpha_1} = \left(x, f'_{\alpha_1}(x)\right)$.

The adversary in this protocol gains no knowledge from the conversations between *A* and *B*. Indeed the adversary could easily simulate these conversations by replacing the encryptions with random strings.

4 PROVABLE SECURITY BASED ON ZERO-KNOWLEDGE

In 1976 Diffie and Hellman presented the first public key agreement system (Diffie-Hellman 1976). In this paper we will use a variant of this which is based on public keys, the Matsumoto-Takashima-Imai (MTI) system (Matsumoto et al. 1986, ISO/IEC Draft Directory 1995). This uses a discrete logarithm setting. Let p be a large prime and g generator of Z_p^*. Each entity i in the network chooses a secret exponent S_i in Z_{p-1} and registers $P_i = a^{S_i} \bmod p$ as its public key with a Trusted Centre (which also chooses the parameters p and g of the system).

When entities A and $\hat{A}$ wish to communicate securely, they first compute the *Diffie-Hellman key* $k_{AB} \equiv (g^{s_A})^{s_B} \equiv (g^{s_B})^{s_A} \equiv g^{s_A s_B} (\bmod p)$. Then, they select secret random exponents R_A, R_B in Z_{p-1} and exchange $X_A = {k_{AB}}^{R_A} \bmod p$ and $X_B = {k_{AB}}^{R_B} \bmod p$, respectively. The common session key is :

$$K_{AB} = {k_{AB}}^{R_A R_B} \bmod p = (X_A)^{R_B} \bmod p = (X_B)^{R_A} \bmod p \ ,$$

which both entities can compute. The protocol is described in Figure 2. Observe that a fresh key is computed each time.

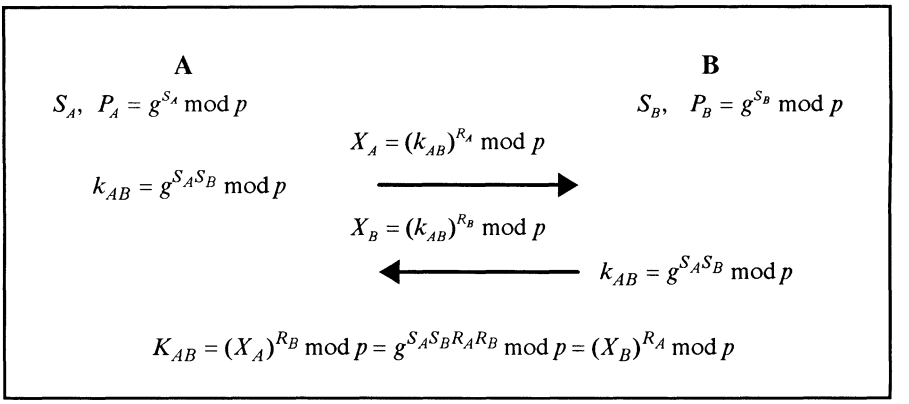

Figure 2 The MTI Key Agreement protocol.

The security of this system against attacks by a passive eavesdropper can be reduced to the difficulty of computing $g^{ab} \bmod p$ given p, g, $g^a \bmod p$ and $g^b \bmod p$, which is known as the *Diffie-Hellman problem**. Indeed suppose that it is easy to compute K_{AB} for a non-negligible fraction of instances given p, g, P_A, P_B, X_A, X_B, and let $g^a \bmod p$ and $g^b \bmod p$ be an instance of the Diffie-Hellman problem. Then, by choosing random exponents S_A, S_B for P_A, P_B and taking $X_A = (g^a)^{S_A S_B} \bmod p$ and $X_B = (g^b)^{S_A S_B} \bmod p$, it is easy to compute $K_{AB} = (g^{ab})^{S_A S_B} \bmod p$ and hence $g^{ab} \bmod p$, for a non-negligible fraction of instances (Adleman et al. 1977)

However the security of the MTI protocol against active adversaries is not proven. The reason for this is that this protocol is not zero-knowledge**. Indeed, an active adversary obtains some knowledge when the session key is revealed. This means that the exchanged messages and the corresponding session keys may be of some use to an adversary who wants to compute a fresh session key. This was not the case for the Bellare-Rogaway protocol for which the session keys were indistinguishable from pseudo-random strings.

We observe that in the MTI system as with all other Diffie-Hellman variants, the role of the Trusted Centre is reduced to managing the public keys of the participating entities.

5 A MODIFIED MTI PROTOCOL

As observed, the MTI protocol is not proven secure because it leaks knowledge which an adversary could possibly use to compute fraudulent session keys. This would be prevented if the entities had to prove to each other that they know their secret keys S_A, $S_{\hat{A}}$ and the random exponents R_A, $R_{\hat{A}}$ respectively (Desmedt-Burmester 1993).

---

* It is not clear if this problem is as hard as the Discrete Logarithm problem, but it is generally regarded as a hard problem.

** In can be shown that if the MTI protocol was zero-knowledge then the Diffie-Hellman problem would be easy. The proof of this is similar to that for the Yacobi protocol in Desmedt-Burmester 1992.

In this paper we propose a simple solution which uses Message Authentication Codes* (MACs) based on zero-knowledge proofs. In order to guarantee that no knowledge about the secret key S_A or the secret exponent R_A leaks, entity A authenticates its message X_A to entity B in the MTI protocol by using a MAC with secret keys: the Diffie-Hellman key k_{AB} and the exponents S_A, R_A.

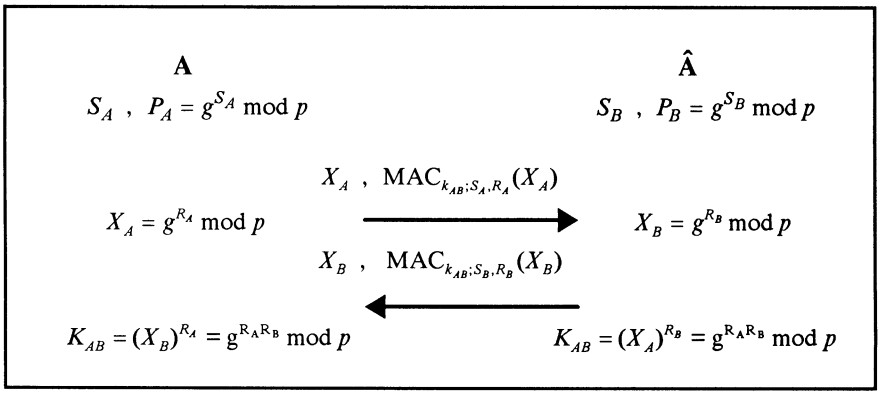

Figure 3 A provable secure variant of the MTI protocol.

Figure 3 describes the proposed modification using zero-knowledge based MACs. Observe that the common session key K_{AB} is independent of the public keys of the entities. In the next section we will show how to implement this protocol efficiently, and discuss its security.

6 AN EFFICIENT IMPLEMENTATION

In the modified protocol (Figure 3) each entity must ensure that no knowledge about its secret key or its secret exponent leaks. To achieve this we use the Schnorr variant (Schnorr 1991) of the zero-knowledge proof of the Discrete Logarithm (Chaum et al. 1988). In this, a Trusted Centre chooses a large prime p, a generator g, a large prime factor q of p-1, and computes $a = g^{p-1/q} (\mathrm{mod}\, p)$. It publishes p, q, a, a security parameter $t < \log_2 q$, and the parameters of a pseudo-random function $f \cdot Z \times Z \times Z \rightarrow$ *

The Schnorr proof of knowledge of the discrete logarithm of $P = a$ p has three interactive steps. In the first step, the Prover selects a random exponent $r \in Z$ and sends to the Verifier. In the second step, the Verifier responds with the random query In the third step, the Prover sends to the Verifier. Then the Verifier checks that , and accepts if it is valid. A fraudulent Prover, who does not know S may try to cheat, i.e. get the Verifier to accept by guessing the query and

* Key dependent hash functions.

finding an appropriate answer. The probability with which the Verifier will accept is called the cheating probability. The cheating probability for this proof is 2^{-t} (Schnorr 1991).

For our implementation we replace the generator g in the modified MTI protocol by $a = g^{p-1/q} \bmod p$ and take all exponents in Z_q (including the secret keys). For a proof of simultaneous knowledge the discrete logarithms of $P_A = a^{S_A} (\bmod p)$ and $X_A = a^{R_A} (\bmod p)$, entity A selects a random exponent $r \in Z_q$ and sends $x = a^r (\bmod p)$ to entity B. B replies with two random queries $e', e \in \{0,\ldots,2^{t-1}\}$. Then, A sends B: $y = r - eS_A - e'R_A (\bmod q)$. The verification is $x = a^y P_A^e X_A^{e'} (\bmod p)$.

To get a MAC we apply the technique proposed by Fiat-Shamir 1987, for obtaining digital signatures based on zero-knowledge proofs. This uses a random function f. The signature of a message m based on the proof of the discrete logarithm of $P = a^S (\bmod p)$ described above, consists of a pair (y,e) such that e is a prefix of $f(m,x)$, where $x = a^y P^e (\bmod p)$. For our modification, we use a pseudo-random function $f_{k_{AB}}$ with key the Diffie-Hellman key k_{AB} shared by the entities A, B. The MAC which authenticates X_A to entity B is $MAC_{k_{AB};S_A,R_A}(X_A) = (y,e,e')$, for which ee' is a prefix of $f_{k_{AB}}(X_A, x)$.

The strength of this type of authentication is that it is essentially zero-knowledge: the knowledge which leaks is limited to the scope of the MAC (Sakurai-Itoh 1993). The security of the proposed protocol is based on the following observations:

- If the entities A and $\hat{A}$ use a zero-knowledge proof of knowledge of the secret keys S_A, S_B and the random exponents R_A, R_B respectively, and if the adversary can compute the session key K_{AB} with non-negligible probability of success, then the Diffie-Hellman problem is easy (the proof is similar to that for the Yacobi 1991 protocol in Desmedt-Burmester 1993).
- If f is a random function and if the MAC can be forged with an adaptive chosen message attack then the Diffie-Hellman problem is easy (the proof is similar to that for signatures in Fiat-Shamir 1987).

In our modified MTI protocol, f is a pseudo-random function. The proof that the second observation remains true when f is replaces by a pseudo-random function is based on the following argument: if an adversary E can forge a MAC when f is pseudo-random, but not when f is random, then E can distinguish pseudo-random functions form random function. This is impossible if E has bounded resources.

Suitable pseudo-random functions for our implementation can be obtained by using the construction proposed by Bellare-Rogaway 1994. This uses the Data Encryption Standard (DES) in Cipher-Block-Chaining mode and a suitable hash function such as the MD5.

7 CONCLUDING REMARKS

In this paper we have considered two security models for key agreement systems. The first one uses probabilistic encryption which makes it impossible for an adversary to distinguish a session key from a random key. With this model a Trusted Centre is needed to select the secret keys for each pair of entities. The second model is based on zero-knowledge. For this model we propose a variant of the Diffie-Hellman key agreement protocol. The main advantages of this protocol are that the Trusted Centre may be offline, and its role is reduced to selecting the parameters and managing the system. The users select their own secret keys and the protocol is efficient, and as secure as the Diffie-Hellman key exchange.

8 REFERENCES

Adleman, L., Manders, K.M. and Miller, G.M. (1977) On taking roots in finite fields, *Annual Symposium on Foundations of Computer Science*, **18**, 175-178.

Alexandris, N., Burmester, M., Chrissikopoulos, V. and Desmedt, Y. (1993) A secure key distribution system, *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography,* Rome, 30-34.

Alexandris, N., Burmester. M., and Chrissikopoulos V. (1992) An efficient public key distribution system, *Proceedings of the IFIP 12th World Computer Congress,* North Holland, 532-539.

Bellare, M. and Rogaway, P. (1994) Entity authentication and key distribution, in *Advances in Cryptology - Crypto 93, Lecture Notes in Computer Science 773* (D.R. Stinson, ed.), Springer-Verlag, 232-249.

Bird, R., Gopal, I., Herzberg, A., Jansen, P., Kutten, S., Molva, R. and Yung, M. (1992) Systematic design of two-party authentication protocols, in *Advances in Cryptology - Crypto '91, Lecture Notes in Computer Science 576* (J. Feigenbaum, ed.), Springer-Verlag, 44 -61.

Blum, M. and Micali, S. (1984) How to generate cryptographically strong sequences of pseudo-random bits, *Siam J. Comput.*, **13**, 850 - 864.

Burmester, M., Desmedt, Y. and Beth, T. (1992) Efficient zero-knowledge distribution schemes for smart cards, *The Computer Journal,* **35**, 21-29.

Burmester, M. (1994) On the risk of opening distributed keys, in *Advances in Cryptology - Crypto '94. Lecture Notes in Computer Science 839* (Y.Desmedt, ed.) , Springer-Verlag, 308-317.

Chaum, D., Evertse, J.H. and Van de Graaf, J. (1988) An improved protocol for demonstrating possession of discrete logarithms and some generalizations, in *Advances in Cryptology - Eurocrypt '87, Lecture Notes in Computer Science 304* (D. Chaum and W.L. Price, eds.), Springer-Verlag, 127-141.

Chrissikopoulos, V. and Peppes, D. (1995) A Practical Conference Key Distribution System, *Information Security-the Next Decade, Proceedings of IFIP/SEC'95, The 11th International Information Security Conference.,* (J. Eloff and S. von Solms eds.), 168-175.

Desmedt, Y. and Burmester, M., (1993) Towards practical proven secure authenticated key distribution, *Proceedings of the 1st ACM Conference on Computer and Communication Security,* Fairfax, Virginia, ACM Press, 228-231.

Diffie, W. and Hellman, M.E. (1976) New directions in cryptography, *IEEE Trans. Inform. Theory*, **IT-22**, 644-654.

Diffie, W., van Oorschot, P.C. and Wiener, M. (1992) Authentication and authenticated key exchanges, *Designs, Codes and Cryptography*, **2**, 107-125.

Fiat, A. and Shamir, A. (1987) How to prove yourself: Practical solutions to identification and signature problems, in *Advances in Cryptology, Proc. of Crypto '86, Lecture Notes in Computer Science 263* (A. Odlyzko, ed.), Springer - Verlag, 186-194.

Goldwasser, S. and Micali, S. (1984) Probabilistic encryption, *Journal of Computer and System Sciences*, **28**, 270-299.

Goldwasser, S., Micali, S. and Rivest, R. (1988) A digital signature scheme secure against adaptive chosen-message attacks, *Siam J. Comput.*, **17**, 281-308.

Gong, L. and Wheeler, D.J. (1990) A matrix key-distribution scheme, *Journal of Cryptology*, **2**, 51-59.

ISO/IEC CD11770-3 (1995) Draft Directory. Information Technology - Security Techniques - Key Management, Part 3: Mechanisms using asymmetric techniques. Key Agreement Mechanism 5.

Kohl, J. (1991) The evaluation of the Kerberos Authentication Service, *EurOpen Conference Proceedings*, 295-313.

Leighton, T. and Micali, S. (1994) Secret-key agreement without public-key cryptography, in *Advances in Cryptology - Crypto 93, Lecture Notes in Computer Science 773* (D. Stinson, ed.), Springer-Verlag, 456-479.

Matsumoto, T., Takashima, Y. and Imai, H. (1986) On seeking smart public key distribution systems, *The Transactions of the IECE of Japan,* **E69**, 99-106.

Needham, R.N. and Schroeder, M.D. (1978) Using encryption for authentication in large networks of computers, *Commun. ACM*, **21**, 993-999.

Sakurai, K. and Itoh, T., (1993) On the discrepancy between serial and parallel of zero-knowledge protocols, in *Advances in Cryptology - Crypto '92, Lecture Notes in Computer Science 740* (E. F. Brickell, ed.), Springer - Verlag, 246-259.

Schnorr, C.P. (1991) Efficient Signature Generation by Smart Cards, *Journal of Cryptology*, **4**, 161-174.

Yacobi, Y. (1991) A key distribution paradox, in *Advances in Cryptology - Crypto '90, Lecture Notes in Computer Science 537* (A. J. Menezes and S. A. Vanstone eds.), Springer - Verlag, 268-273.

Yao, A.C. (1982) Theory and applications of trapdoor functions, in *23th Annual Symp. on Foundations of Computer Science (FOCS)*, IEEE Computer Society Press, 80-91.

9 BIOGRAPHIES

Nikos Alexandris is a Professor in the Department of Informatics at the University of Piraeus. He received his BSc from Athens University and his PhD from UMIST in 1978. His

research interests include Expert Systems, Information Security and Cryptography. He is a member of the Greek Mathematical Society, Greek Computer Society and BCS.

Mike Burmester is in the Information Security Group at Royal Holloway (University of London). He received his BSc from Athens University and his PhD from Rome University in 1965. Since 1990 he has been working in Cryptography and Data Security. He is a member of the Greek Mathematical Society, Greek Computer Society and International Association for Cryptographic Research.

Vassilios Chrissikopoulos is an Associate Professor in the Department of Informatics at the University of Piraeus. He received his BSc from the University of Thessaloniki and his PhD from Royal Holloway, University of London in 1983. His research interests include Expert Systems, Information Security and Cryptography. He is a member of the Greek Mathematical Society, Greek Computer Society, Operational Research Society of the UK and BCS.

Dimitrios Peppes is a PhD Student in the Department of Informatics at the University of Piraeus. He received his MSc at Queen Mary and Westfield College, University of London in 1990. His current interests are in the areas of Applied Cryptography, Network Security and Database Security. He is a member of the Greek Computer Society.