R. Merckling (HP)
A. Anderson (HP)
March 1994

DCE SMART CARD INTEGRATION

1. INTRODUCTION

Smart cards represent a new technology that has tremendous potential for enhancing the security of distributed systems. This RFC, 57.1, introduces a high-level specification for DCE usage of smart cards as part of authentication. A predecessor RFC, titled ‘‘Smart Card Introduction’’ [MER 57.0], explains what smart cards are and why DCE would want to make use of them.

The goal of this RFC is to show how to integrate smart card technology with minimal impact on DCE protocols, while gaining some significant advantages in additional security. The solution described, although developed for only the DCE login application, does not preclude multi-application use of the smart cards in the future.

2. TERMINOLOGY

(a) *Account*: A security database entry associating a principal with a key and other security-related information such as privileges.

(b) *Dedicated file*: A smart card file containing file control information and, optionally, memory available for allocation. It may be the parent of elementary files and/or dedicated files [ISO 7816-4]; roughly, a smart card ‘‘file directory’’.

(c) *Elementary file*: A set of data units or records in the smart card that share the same identifier [ISO 7816-4]; roughly, a smart card ‘‘file’’.

(d) *Key*: A parameter used in conjunction with a cryptographic algorithm that determines:

   (i) The transformation of plaintext data into ciphertext data,

   (ii) The transformation of ciphertext data into plaintext data, or

   (iii) A message authentication code.

(e) *Long-term key*: A key that is used over a relatively long period of time. Compare with ‘‘session key’’. In DCE, each principal has a unique long-term key that is used in authenticating a user or server claiming to be that principal. For interactive principals — ‘‘users’’ — the long-term DCE key is derived from the user’s password.

(f) *Master file*: The mandatory unique dedicated file representing the root of the smart card file structure [ISO 7816-4].

(g) *Message authentication code*: A cryptographic checksum.

(h) *Passivation layer*: A layer of dielectric material covering the circuitry on a smart card that protects the chip from impurities and dust and prevents passage of radiation associated with probes, such as electron-beam microscopy. Often referred to as a ‘‘seal’’.

(i) *Password*: A sequence of alphanumeric and punctuation characters entered by a user to authenticate to a computer system (including to a smart card).

(j) *Principal*: An entity that is capable of believing that it can communicate securely with another entity. In DCE Security, principals are represented as entries in the Registry database. DCE principals include the following [OSF 93]:

   (i) Users, who are also referred to as ‘‘interactive principals’’.

(ii) Instances of DCE servers.

(iii) Instances of application servers.

(iv) Computers in a DCE cell.

(v) Authentication Service surrogates.

(k) *Session key*: A random number generated to serve as a key for a specific transaction or set of transactions.

(l) *Smart card*: A credit card-sized, tamper-resistant security device that relies on VLSI chip technology for information storage and information processing.

3. TWO-FACTOR AUTHENTICATION PROCESS

The major weakness in DCE/Kerberos security is the reliance on passwords for the authentication of interactive principals. Passwords need to be reasonably easy to remember, or users will write them down, introducing yet another security vulnerability. Yet easy-to-remember passwords use only a small fraction of the DES key space, and are subject to dictionary attacks.

Smart cards can dramatically reduce password exposure by supporting a 2-factor, 2-step authentication mechanism [KOHL 93], [KRAJ 93].

With passwords only, a user must know one piece of secret information — the valid password for a principal name — in order to log in to a DCE cell. There is one step involved in authentication — supply a login name and its password to the system.

With smart cards, a user must possess a unique physical token — the smart card — as well as knowing one piece of secret information — a password associated with the card. Knowledge of the card password is of no use without possession of the card itself. Possession of the card is of no use without knowledge of the associated password.

With passwords only, there is one step to the authentication process — the user principal is authenticated to DCE. With smart cards, there is an additional step involved — before being authenticated to DCE, the user principal must be authenticated to the smart card. The first step depends on knowing the correct password. The second step depends on possessing the correct smart card — the one containing the user's long-term DCE key.

We propose using the smart card's secure file system to store the user's long-term DCE key. To access the key and log in to DCE, a user must possess the card and must also know the password required to access the key file on the card.

The details of this solution are contained in the following sections.

4. CARD REQUIREMENTS

To achieve the additional security provided by our solution, cards must meet certain minimum physical and functional requirements.

4.1. Physical Access Requirements

The card must conform to the accepted International Standards [ISO 7816-1], [ISO 7816-2], and [ISO 7816-3]. Failure to conform requires a greater effort in porting the DCE interface, requires the card vendor to prove independently that the low-level security characteristics of the card meet the abstract requirements of the DCE model, and requires the card purchaser to verify the proof.

The card must be tamper-resistant. Common tamper-resistance features include: chip enclosed in a hard, opaque, tamper-evident sealing coating over a passivation layer; circuitry to disable card output if the passivation layer is removed or if temperature, voltage, or frequency fluctuates outside the specified operating range.

Cards with higher levels of tamper-resistance generally have a higher cost. The card issuer must evaluate with the card vendor the appropriate level of tamper-resistance for the issuer's application.

Failure to provide adequate tamper-resistance makes the long-term key stored on the card vulnerable to physical attacks, reducing security when a card is lost or stolen.

4.2. File System Requirements

The card operating system must support a file system capable of storing at least six files:

(a) Long term key file containing a DES key (an 8-byte value, of which only 56 bits are cryptographically ‘‘active’’).

(b) Login name file containing an ASCII principal identifier of up to 2054 bytes (/.../<1024-byte cell name>/<1024-byte login name>).

(c) Architectural model number file containing an ASCII string of up to 5 bytes.

(d) Password expiration file containing a 6-byte ASCII string.

(e) Password lifetime file containing a 32-bit binary value.

(f) Password strength file containing a 32-bit binary value.

The file system must support external operations to read and write the long term key and password expiration files, and to read the login name, password expiration, and password strength files throughout the active usage lifetime of the card. Failure to conform to this specification means the card is unable to support the functionality required for our solution.

4.3. Access Control Requirements

The card file system must provide access controls that support the use of a password to protect access to the file containing the DES key. Passwords of up to 8 bytes in length must be supported. Without presentation to the card operating system of the password associated with the DES key file, there should be no access to the DES key stored in the card file system. Failure to conform to this specification negates the additional security that smart cards offer.

The card must support detection of sequential invalid access attempts (a request for a protected card operation without supplying the correct password for the operation). Upon detecting N sequential invalid access attempts on any operation (where N is either configurable, or is a fixed number recommended to be no greater than 8), the card must deny all further access to all operations except for possibly one ‘‘unlock/erase’’ operation. This one operation, if supported, must erase the DES key stored in the card file system before restoring normal access to the card. Failure to conform to this specification leaves a lost or stolen card vulnerable to off-line dictionary password-guessing attacks. Accounts associated with lost or stolen cards are subject to compromise until an administrator invalidates the account in the DCE Registry.

The card file system must support access controls that allow only the user to modify the password associated with the DES key file (with the possible exception described in the next paragraph).

An interesting implementation would have a way of providing administrative access to unlock/erase a locked card (see above), set a new DES key on the card and in the DCE Security Registry, set a new card access password, etc., but that is a vendor-issue not required for the basic functionality specified in this RFC.

5. DCE SMART CARD IMPLEMENTATION

5.1. Card Preparation

Card preparation involves card operating system operations that have not yet been standardized. The operations are defined in terms of general characteristics of available cards.

The following notation is used for access control permissions:

```
Perm    Semantics
----    -------------------------------------------------------
R       Contents readable.
W       Contents writable.
A       Additional elementary files and distinguished files may
        be added.
U       File may NOT be deleted.
*/PW    Operation permitted only after initial presentation of
        user's password.
*/O     Operation is permitted, but not required.
```

Cards used with DCE must be customized with the following file and access control structure. Absence of a permission indicates that the permission is NOT allowed. Uses of these files is described under ‘‘Changes to DCE Protocols and Interfaces’’.

```
Master File: root
Access: R, U, A/O

Elementary File: /root/username
Access: R, U, W/O/PW

Elementary File: /root/arch_model
Access: R, U

Elementary File: /root/pw_expiration
Access: R, U, W/PW

Elementary File: /root/pw_lifetime
Access: R, U, W/O/PW

Elementary File: /root/pw_strength
Access: R, U, W/O/PW

Distinguished File: /root/DCE
Access: R/PW, U, A/O/PW

Elementary File: /root/DCE/key
Access: R/PW, W/PW, U
```

The contents of the `arch_model` file must be unique for a given ‘‘solution model’’ [MER 57.0], i.e., for a given card file directory structure and card behavior model. For the structure and behavior described in this RFC, the `arch_model` must be `1.0`.

5.2. Changes to the DCE Registry Structures

A well-known ERA boolean schema entry named `sc_account` is defined. Instances of this attribute are attached to principal nodes in the registry database to indicate whether the account requires use of a smart card or not.

5.3. Changes to DCE Protocols and Interfaces

Smart cards affect DCE protocols and interfaces in the following ways. The descriptions assume that DCE 1.1 Preauthentication [PATO 93] is supported by the DCE Security Server and by all client machines where smart cards are used.

5.3.1. DCE login process

The login information flow can be diagramed as follows:

```
          <username, card password>
1a.  USER ---------------------------------------> HOST

          <user account information>
1b.  HOST <---------------------------------------  LOCAL REGISTRY

          <TGT request using card password>
1c.* HOST ---------------------------------------> SECURITY SERVER

          <error, ``smart card account''>
1d.* HOST <---------------------------------------- SECURITY SERVER

          <``smart card account''>
1e.* HOST ---------------------------------------> LOCAL REGISTRY

          <username file, arch_model file>
2a.  HOST <---------------------------------------- CARD

          <card password>
2b.  HOST ---------------------------------------> CARD

          <long-term DCE key>
2c.  HOST <---------------------------------------- CARD

          <TGT request using long-term DCE key>
3a.  HOST ---------------------------------------> SECURITY SERVER

          <DCE credentials>
3b.  HOST <---------------------------------------- SECURITY SERVER

(* = Step omitted if Local Registry (step 1b) indicates user has
     a smart card account.)
```

Each step in the smart card authentication process is described in the following sections.

The required changes to DCE are completely hidden inside the `sec_login_*` API's. There are no changes to external DCE API's.

5.3.1.1. Initial TGT request

As in the normal login process, the smart card user enters first a username and then a `password` at the prompting of some login utility (step 1a).

Prior to generating an initial TGT request, the `sec_login_*` API accesses the Local Registry for cached information regarding this user's account (step 1b). If such information is found, and if the account is flagged as a ''smart card account'', then `sec_login_*` immediately skips to the step (step 2a) that attempts to access the smart card.

Except as described above, the steps for requesting the initial TGT (KRB_AS_REQ) are not changed on the client side: a normal pre-authenticated request [PATO 93] will be generated using the key derived from the password supplied by the user to the login utility (step 1c).

The steps for processing a KRB_AS_REQ by the KDC are unchanged unless there is an error in decrypting the timestamp using the random-key. In this case, the KDC queries the Registry Service for the value of the principal's `sc_account` attribute. If this attribute exists and has the value `TRUE`, then the KDC returns the current error value, but includes the value of the `sc_account` attribute in the error information parameter (step 1d).

Back on the client side, when the KRB_AS_REP returns an error status, the client checks the error information parameter. If it indicates that the `sc_account` attribute is `TRUE`, then the client flags the user's account in the

Local Registry as a smart card account (step 1e). Otherwise, the existing error is returned to the login utility.

5.3.1.2. Authentication by the smart card

Having detected that this user is a smart card user, the `sec_login_*` API now attempts to read the `/root/username` and `/root/arch_model` files from the smart card (step 2a). If the files can't be read (e.g., no reader attached, or no smart card inserted in reader), the `sec_login_*` API returns error ''Unable to read smart card'' and exits to the login utility.

If the returned `username` value does not match the login name of the user, then the client code reports the existing login failure message to the user, and resumes with the existing code path, although detailed information about the failure may be reported to the DCE Audit facility.

If the returned `arch_model` value is not `1.0`, then the client code returns error ''Incompatible smart card version'' and exits to the login utility.

If the value of the `/root/username` file can be read, and if the value matches the login name of the user, then the client code attempts to read the DES key from the `/root/DCE/key` file, using the password supplied by the user to gain access to the card (step 2b).

If this attempt is unsuccessful, then the client reports the existing login failure message to the user, and resumes with the existing code path, although detailed information about the failure may be reported to the DCE Audit facility.

5.3.1.3. Authentication by the DCE security registry

If the attempt to access the DES key file on the card is successful (step 2c), then the steps for requesting a TGT (KRB_AS_REQ) are repeated, except that the DES key value obtained from the smart card is used throughout the protocol instead of the user's password (steps 3a and 3b).

5.3.2. Account creation

A new field, `sc_account` is created as part of the standard principal registry information. The value is boolean.

`rgy_edit` queries the administrator for a value for this field as follows:

```
rgy_edit=> add
Add Account=> Enter account id [pname]: <user name>
Enter account group [gname]: <group>
Enter account organization [oname]: <organization>
Smart card account [y/n]? (n) y
Enter password for user card access: <user password>
Retype password for user card access: <user password>
Enter your password:
Enter misc in: ()
Enter home directory: (/)
Enter shell: ()
Account valid [y/n]? (y)
Enter account expiration date [yy/mm/dd or 'none']: (none)
...
Create/Change auth policy for this acct [y/n]? (n)
...
Insert a blank smart card into card reader, then press RETURN
```

In the case illustrated above where the administrator is creating a smart card account, the data structures described in the ''Card Preparation'' section are created on the smart card at this point, and are initialized with the appropriate values (`arch_model` is `1.0` for this RFC). Next, a random key is generated and written to the long term key file on the card. The random key is then provided as input to the standard account creation API. The ''user password for card access'' is used only in initializing the DCE authorization data on the smart card. It is not stored in the DCE Registry.

5.3.3. Account modification

The procedures for account modification change in a way similar to those for new account creation. If the account is currently a smart card account, the administrator or account owner is asked whether the account should remain a smart card account. If the account is not currently a smart card account, the administrator or account owner is asked whether the account should become a smart card account.

For the case of an existing smart card account being updated, but remaining a smart card account, the user is prompted to insert the smart card (not blank!) into the reader, and appropriate modifications are written.

For the case of an existing smart card account ceasing to be a smart card account, a warning is displayed that proceeding will invalidate the smart card. If the user continues, the user is forced to enter a new password, which now becomes the base for the user's long term key stored in the DCE Registry.

For the case of a non-smart card account being converted to a smart card account, all information required for the card is requested and written to the card as described above for new accounts. If the card in the reader has already been initialized, then the login name is checked and the password supplied is used in an attempt to read the long term key file from the card. If both succeed, then the existing card is updated (this would be the case where a smart card account was temporarily converted to a non-smart card account).

5.3.4. Password and key modification

Card access password modification can be done either through `rgy_edit` (via the `change account` operation), or through `kpasswd`. In the case of `kpasswd`, the DCE Registry is not involved. In the case of `rgy_edit`, the DCE Registry is not involved in the password changing, although other account information being changed may involve the Registry.

The login process forces change of card access passwords if the date information stored in the `/root/pw_expiration` file on the card indicates that the current card access password has expired. Proposed passwords are checked using the requirements encoded in the `/root/pw_strength` file (encoding to be specified in Functional Specification). Once the password has been changed, the new `/root/pw_expiration` file value is computed and written using the current date plus the number of days in the `pw_lifetime` file.

Long term keys for smart card accounts may be updated explicitly using the `rgy_edit change account` operation, where generation of a new random password is requested.

Long term keys for smart card accounts are also updated automatically as part of the login process whenever the long term key is within two weeks of expiration.

Change of long term key, either via `rgy_edit` or automatically, requires a two-phase commit protocol in order that the key stored in the card does not get out of synch with the key stored in the DCE Registry. This protocol will be designed using the existing protocol for replica updates as a base.

5.4. Card Administration

To avoid a ‘‘chicken and egg problem’’, there must be at least one administrator (i.e., a principal with `sec_admin` privileges) for the cell who does not have a smart card account when the DCE Security Server is first configured. This is because some administrator must be able to log in in order to create smart card accounts for other users. Non-smart card administrator accounts may be changed to smart card accounts once the DCE Security Server has been configured.

Administrator actions may be required in the following situations. Unless indicated otherwise, administrator actions may be performed only when the administrator has physical possession of the smart card — the operations may not be performed on behalf of a smart card user at a remote location.

5.4.1. Creation of new smart card accounts

Only an administrator can create a new account. This is done via the `rgy_edit account add` operation described above.

5.4.2. Unlocking a card

If the implementation supports administrator unlock (requires a vendor-specific ‘‘unlock/erase’’ operation), then an administrator can unlock a card for a user who has exceeded the allowed number of invalid access attempts. This is done using the `rgy_edit unlock account` operation. This is a vendor-specific operation and is not described in this RFC.

5.4.3. Resetting card access password

If the implementation supports administrator unlock (see above), then an administrator can unlock and reset the card access password (and long term key) for users who have forgotten their password. This is also done using the `rgy_edit unlock account` operation. This is a vendor-specific operation and is not described in this RFC.

5.4.4. Invalidating smart card accounts

When a user reports a lost or stolen smart card, an administrator should immediately invalidate the account using the `rgy_edit change account` operation. This operation can be done without possession of the smart card — it requires only marking the account invalid in the DCE Registry, or converting the account to a non-smart card account (see above).

Until the administrator issues the user a new smart card, or until the administrator converts the account to a non-smart card account, the user will be unable to log into the computer system (note that this is a strong motivation to report loss or theft promptly).

Smart card accounts should be invalidated by deleting the account when a user leaves the company, moves out of the cell’s administrative domain, or otherwise ceases to be an authorized user. Account deletion does not require administrator possession of the smart card.

5.5. Policies

This section describes the policies for use and administration of cards that are required to support card security. Although some of these policies can be enforced by the DCE Security software, each organization must educate its employees regarding what the policies are and why they exist. Employees must be motivated to follow organization policies for them to be fully effective.

5.5.1. Password change interval

Card access passwords must be changed at regular intervals. If the interval causes too much inconvenience, however, then users will be motivated to circumvent or subvert the policy, such as by writing down passwords or by re-using passwords.

We suggest use of the SOSPG issued by the X/OPEN EGSEC 93 [EGSEC 93] as a reference in selecting change interval policies.

The change interval is enforced using the password lifetime policy specified with the `rgy_edit policy` operation. The value in effect at the time the smart card account is created is stored in a protected file on the card. The `sec_login_*` software requires the user to change passwords when the current password exceeds the specified lifetime.

5.5.2. Password strength

Card access passwords should not be easily guessable within the maximum invalid access attempts enforced by the card. An organization should specify policies for password strength that minimize guessing.

We suggest use of the SOSPG issued by the X/OPEN EGSEC 93 [EGSEC 93] as a reference in selecting password strength policies.

Policies on password length and limited policies on character variety can be specified using the `rgy_edit policy` command. An encoded form of the policies in effect at the time the smart card account is created is written to a protected file `/root/pw_strength` on the smart card. The `sec_login_*` and `kpasswd` commands enforce these policies when the user changes the card access password.

5.5.3. Card access password storage

The most common way passwords are compromised is when users write them down, especially when the written copy is kept with the card or with the machine where the user logs in.

Organizations must have a policy that passwords are not to be written down or shared with other individuals.

5.5.4. Card loss or theft

Organizations must have a policy that lost or stolen smart cards are to be reported to an administrator promptly. The method for contacting an appropriate administrator must be well publicized.

Administrators must be trained to respond appropriately to card loss or theft reports: the account should be invalidated immediately using the `rgy_edit change` operation. If the organization allows an administrator to change a smart card account to a non-smart card account until a lost or stolen card is replaced, then there must be well-understood procedures for verifying that the loss report comes from the authorized user and for delivering the new account password to the authorized user.

5.5.5. Invalid accounts

Accounts of individuals who cease to be authorized users should be deleted or invalidated immediately.

5.5.6. Smart card management

Organizations must ensure that the procedures for ordering, numbering, and storing un-issued smart cards are secure.

5.5.7. Smart card distribution

Organizations must ensure that the procedures for creating smart card accounts, and for delivering smart cards and their initial card access passwords to users are secure.

5.6. Compatibility

The `arch_model` file in the master file must be unique for a given card file directory structure and card behavior. In order to allow future enhancements to smart card protocols to be implemented, the client side security runtime software must know which smart card architectural models it supports. It is expected that client side security runtime software implementing new smart card architectural models will continue to support all previous models, except where prohibited by cell policy.

6. SECURITY ASSUMPTIONS

This section describes the overall security assumptions implied by the solution presented above.

6.1. Trust Assumptions

(a) The client host machine OS is trusted to limit access to the data buffers read from and written to the smart card to authorized processes. Unauthorized processes should not be able to access the long-term key or the user's password by snooping data buffers. This is a standard DCE Security assumption.

(b) The client host machine OS is trusted also to limit access to the file system to authorized processes. Unauthorized processes should not be able to access credentials files or key tables. This is a standard DCE Security assumption.

(c) The client host machine OS is also trusted to protect the memory assigned to the login utility and used by the `sec_login_*` API's for temporary storage of passwords and keys. This is a standard DCE Security assumption.

(d) The login utilities and the `sec_login_*` API implementation on the client host machine must be trusted not to disclose the user's password, long-term key, or DCE credentials. This is a standard DCE Security assumption.

(e) The `sec_clientd` implementation running on the client host machine must be trusted not to disclose the user's password, long-term key, or DCE credentials. This is a standard DCE Security assumption.

(f) The DCE Security Server machine is assumed to be trusted once it has been certified to have possession of the client host machine principal's key (certification is done by the client host machine's `sec_login_valid_and_cert_ident` API). This is a standard DCE Security assumption.

(g) The physical connection between the user's keyboard and the client host machine must be trusted. Unauthorized users should not be able to tap the connection to read the user's password as it is entered. This is a standard DCE Security assumption.

(h) The physical connection between the smart card reader and the client host machine must be trusted. Unauthorized users should not be able to tap the connection to read the user's password or the user's long-term key.

(i) The smart card reader itself must be trusted not to replay information copied from a previous transaction, or to disclose the user's password or the user's long-term key.

6.2. Non-Trust Assumptions

(a) The network connection between the user's host machine and the DCE Security Server machine is not assumed to be trusted. All security-relevant information in the protocols is cryptographically protected when sent across the wire.

7. ACKNOWLEDGMENTS

The following members of the HP Chelmsford System Software Lab were significant contributors to this RFC: Maryann Hondo, Sean Mullan, and Joe Pato.

Gilles Lisimaque reviewed an early draft, providing extensive helpful comments and corrections. Mr. Lisimaque is a member of ISO/IEC JTC1/SC17/WG4, a member of ANSI X3B10.1, and is cochairman of the Technical Committee of the Smart Card Forum. Mr. Lisimaque is one of the founders of Gemplus Card International and Executive Vice President of Gemplus Card International Corp, Gaithersburg, MD.

REFERENCES

[EGSEC 93] X/OPEN, ''Secure Open Systems Procurement Guide (SOSPG)'', X/OPEN EGSEC, December 1992.

[ISO 7816-1] ISO, ''IS 7816-1: Identification cards — Integrated circuit(s) cards with contacts — Part 1: Physical Characteristics'', 1987.

[ISO 7816-2] ISO, ''IS 7816-2: Identification cards — Integrated circuit(s) cards with contacts — Part 2: Dimensions and Locations of Contacts'', 1988.

[ISO 7816-3] ISO/IEC, ''IS 7816-3: Identification cards — Integrated circuit(s) cards with contacts — Part 3: Electronic signals and transmission protocols'', 1989.

[ISO 7816-3.1] ISO/IEC, ''IS 7816-3, Amendment 1: 1989, Identification cards — Integrated circuit(s) cards with contacts — Clause 9 to be inserted in part 3: Protocol type T=1, asynchronous half duplex block transmission protocol'', December 1992.

[ISO 7816-3.2] ISO/IEC, ''7816-3, Amendment 2: 1994, Identification cards — Integrated circuit(s) cards with contacts — Revision of protocol type selection''. Not published.

[ISO 7816-4] ISO/IEC, ''CD 7816-4.2: Information Technology — Identification Cards. Integrated Circuit(s) Cards with Contacts, Part 4, Inter-industry commands for Interchange'', CD ballot 7186-4.2 circulated for vote by SC17 on 12/14/93. Also available for ANSI members under the reference X3B10 94/562.

[KOHL 93] J. Kohl and C. Neumann, ''The Kerberos Network Authentication Service'', Internet Engineering Task Force, Networking Group, Internet Draft RFC 1510, Sep 1993.

[KRAJ 93] Marjan Krajewski, Jr., "Smart Card Augmentation of Kerberos", Proceedings of the Privacy and Security Research Group Workshop on Network and Distributed System Security, February 11-12, 1993, San Diego, CA, pp.119-123.

[MER 57.0] Roger Merckling and Anne Anderson, ‘‘OSF/DCE SIG RFC 57.0: Smart Card Introduction’’, March 1994.

[NIST 140] NIST, ‘‘FIPS DRAFT 140-1 — Security Requirements for Cryptographic Modules’’, April 1991.

[OSF 93] OSF, ‘‘OSF DCE Application Development Guide — Part 6. DCE Security Service — Chap 40. Authentication’’, Revision 1.0, Prentice Hall, Dec 1993.

[PATO 93] Joe Pato, ‘‘OSF/DCE SIG RFC 26.0: Use of pre-authentication to avoid password guessing attacks’’, June 1993.

AUTHORS’ ADDRESSES

Roger Merckling
Hewlett-Packard France
38053 Grenoble Cedex 9
Etablissement de Grenoble
5, avenue Raymond Chanas
38320 EYBENS
France
Internet email: Roger_Merckling@hp6300.desk.hp.com
Telephone: 33-76-62-1131

Anne Anderson
Hewlett-Packard Company
300 Apollo Drive
Chelmsford, MA 01824
USA
Internet email: aha@ch.hp.com
Telephone: 1-508-436-5707