Open Software Foundation
Request For Comments: 71.0
February 1995
G. Gaskell (DSTC)
M. Warner (Telstra)

IMPROVED SECURITY FOR SMART CARD USE IN DCE

1. INTRODUCTION

Organisations are becoming increasingly aware of computer security. Some large corporations (including Telstra) have mandated the use of token-based authentication for all new applications. It is desirable to strengthen the authentication in DCE. Instead of only relying on password authentication of users, authentication should require proof that the user possesses a unique token. The use of a biometric characteristic of the user is also desirable. Smart cards are seen as an ideal token technology. If DCE is to satisfy the security policies of these organisations, smart card authentication will have to be supported.

This RFC examines the use of smart cards conformant with ISO International Standard 7816 to provide strong authentication in Kerberos based authentication in DCE. Trade-offs between card functionality and capacity on one hand, and security and user convenience on the other are examined. This RFC only requires the use of technology that is currently available.

This RFC does not consider the use of smart cards with public key based authentication. The authors are currently preparing another RFC which will deal with public key based authentication in general, making use of smart cards to exploit fully the architectural advantages available when public key cryptography is used.

1.1. Current Login Procedure

The current Kerberos login facility (`kinit` in MIT Kerberos, `dce_login` in DCE) acts as an intermediary between the human user and the authentication server. The user is prompted for a user name. This is then sent to the authentication server, which responds with a ticket granting ticket (TGT) encrypted with the user's key. The login facility then prompts for a password. The user's key is derived from the password entered by the user, and, if this successfully decrypts the TGT, the TGT may be used to request further services. After this point, the user's original key is no longer used, since a session key is issued with the TGT. Note that the TGT and accompanying session key have a limited lifetime.

In DCE this initial exchange is followed by a call to the privilege server to obtain a Privilege Ticket Granting Ticket (PTGT) which includes the user's Privilege Attribute Certificate (PAC).

1.2. Limitations of Kerberos

The Kerberos authentication system provides a considerable improvement over traditional password-based systems in the way authentication information is exchanged between claimants and verifiers. Nevertheless, the secret keys, on which all security ultimately depends, are derived via a one-way function from a password. Such passwords must be remembered by humans, and, as such, are often poorly chosen [GONG]. The key space resulting from the use of human passwords is often a small fraction of the total possible key space. Thus, Kerberos is vulnerable to password guessing attacks.[1]

Another threat which Kerberos does not protect against is that posed by a compromised workstation. When a user attempts to log in to such a workstation, it could capture the user's password, and hence the user's secret key, thereby impersonating the user until her password is changed.

---

1. There have been proposals to reduce the threat of off-line password guessing by requiring a pre-authentication phase. See the proposal of [PATO], implemented in DCE 1.1, noting the dependency on secure ''machine'' keys it introduces.

2. FUNDAMENTAL OBJECTIVES OF SMART CARD INTEGRATION

The problems described above can be greatly reduced by incorporating smart cards into DCE. Keys used in the authentication exchange can be derived from the whole key space, with no concessions to human memory, since they can be stored on the card. This will dramatically increase the expected time required for an exhaustive key search, in general making such an attack infeasible.

By performing at least some decryption on the smart card, cryptographic keys need not be exposed to the local host. This reduces the threat of a corrupt workstation gaining the user's secret keys. This is most critical in the case of long term user keys, which are currently derived from user passwords. Session keys, which have a limited lifetime, should also be protected. However, it may be acceptable to expose these keys to the local workstation which the user logs into, since this workstation has been deemed trustworthy enough to act on behalf of the user.

3. INTEGRATION OPTIONS AND ISSUES

3.1. Possible Approaches to Integration

There are a number of possible approaches to using smart cards with Kerberos authentication. What distinguishes each of these approaches from the others is the amount of processing performed by the smart card, and hence the level of exposure of cryptographic keys. A number of approaches are outlined below.

The approaches examined below are in order of increasing smart card processing complexity:

(a) Smart card as a key store only (3.1.1).

(b) Decryption of the TGT performed by the smart card (3.1.2).

(c) Service ticket decryption by the smart card (3.1.3).

(d) All decryption by the smart card (3.1.4).

The use of smart cards should simplify the login procedure. Where multiple authentication and naming domains exist, it should be possible (in theory) to log in from remote domains, assuming that the necessary inter-realm keys have been exchanged previously.

3.1.1. Key store

The smart card can be used as a simple key store. This is the approach described in [MERK]. This requires a very simple memory card, without any on-card processing capabilities.

While this solution eliminates the problems of password-derived keys, it does nothing to counter the threats posed by a corrupt workstation. Exposure of the user's master key to the workstation is a very real security threat, since too often it is simple to replace the login command with a version which records the user's key [KIM].

3.1.2. TGT decryption

By choosing a smart card with sufficient on-card processing to perform decryption, the user's long term key need never be exposed to the workstation. Instead, the encrypted TGT is passed to the smart card where it is decrypted to yield the session key, which is returned to the workstation. The session key has a limited lifetime — typically 8–10 hours — and during this period, the workstation may issue requests on behalf of the user. During this time, a corrupt workstation could issue fraudulent requests to any server to which the user has access rights.

3.1.3. Service ticket decryption

The session key could also be kept secret from the workstation. Under this approach, the smart card would encrypt requests to the authentication server for tickets to use individual servers, and decrypt the returned tickets. The session keys for individual services would then be provided to the workstation. In this way, a corrupt workstation could only invoke requests on the services for which it obtains a ticket, and not obtain tickets for any other services it chooses. Thus if a user logs on to check her mail for example, the corrupt workstation may obtain access to her mailbox, but not to other services. Of course, this assumes that the smart card only performs encryption (authenticator generation) for service requests which the legitimate user wishes to invoke.

The smart card must be able to detect fraudulent requests generated by the corrupt workstation if any additional security is to be provided over the previous alternative. To do this it must be able to inform the user of the request which it has been asked to sign and obtain an authorisation to do so. This requires a separate secure communication path from the card to the user which bypasses the workstation (and possibly the reader if this is not tamper-proof). This communication path would be used before each new service is invoked for the first time.

3.1.4. All cryptography

All cryptographic processing could be performed on the smart card, without any keys being exposed to the workstation. This approach is described in [KRAJ]. In theory, this would prevent the corrupt workstation from issuing any unauthorised requests. In practice, the authentication problem is simply shifted from the server to the smart card, which must determine whether requests are genuine or not.

Again, a separate communication path between the user and her card is required, and the user must interact with the card before each individual request is encrypted. This implies an unacceptable overhead, in terms of both delay and the additional human activity involved in entering PINs, etc., into the card. Without such a mechanism, the card would sign any request the workstation generates, which would provide no additional security over the second alternative.

3.2. Preferred Approach

Of the four alternatives described above, the first does nothing to protect against a corrupt workstation. The third and fourth alternatives require frequent human interaction, as well as a secure communication path between the user and the card. This would require a more expensive card, as well as introduce unacceptable complexity for human users. Thus, the second alternative is recommended.

3.3. Smart Card Activation

To prevent lost or stolen cards being used by an attacker, the smart card should require a personal identification number (PIN) to be entered before it will decrypt the TGT. Ideally, this PIN should be entered directly into the card, to prevent its exposure to the local host or card reader. Unfortunately key pads on smart cards are expensive, and also reduce the reliability of the cards.

Alternative solutions are to enter the PIN via the host or the smart card reader. The host is by far the easier solution, but results in the greatest exposure of the PIN. Entering the PIN into the smart card reader provides an intermediary solution. However, this obviously requires a key pad and display on the card reader. Such a card reader would be considerably more expensive than a reader without such key pad and display.

In general, entering PINs via the local host provides adequate security, since the PIN itself is of no value without the card. This prevents the misuse of lost cards, and ensures that an attacker will have to capture both the card and PIN.

The system should allow different types of card reader to be used, and fully support available functionality (e.g., displays, key pads).

3.4. Smart Card Technology Required

The two main areas of functionality required of the card are memory and processing capability. We have already indicated that the card must be able to decrypt the TGT. In DCE, the DES algorithm is normally used for all encryption. The login process should not take more than a few seconds. Since the length of Kerberos tickets is of the order of two hundred bytes, the card must be able to decrypt at around 100 bytes per second. In addition, communication delays will affect the response time. Given the normal baud rates and communication overheads, communication delays may be of the order of two seconds.

The data to be stored on the card should include the user's login name, the name of the cell within which the user is registered and the user's key. Cell and user names are of variable length, but, given that they should be universally unique, the combined length may run to about 100 bytes (typically it will be less). This represents the amount of free non-volatile memory required in the card (in addition to program and card operating system memory).

An example of a suitable smart card is the Gemplus MCOS 24K DES card, which includes 3K bytes (24K bits) of EEPROM, 128 bytes of RAM and an on-card DES function.

4. OVERVIEW OF SMART CARD SYSTEM DESIGN

Given the approach described above, it is possible to develop many different implementations. In general, these implementations would be unable to work together, in the sense that a smart card from one implementation could not be used with the reader and login script from another. It may be desirable to allow users to log in to ‘‘foreign’’ systems in order to gain access to their ‘‘home’’ services, for example to check their mail. It is therefore essential to standardise the interface from the card to the DCE security services. Standard smart card commands should be defined by OSF for decryption of the Kerberos TGT. The ISO command format specified in DIS 7816-4 [ISO] should be used.

Note that this design assumes that entire DCE cells adopt the smart card system, in which case no modifications are required to the authentication server or DCE registry. It would still be possible for users without a smart card account to log in at a host with a smart card reader (if this is allowed by security policy) by supporting the original password based login script in addition to the smart card login script. Thus smart card authentication can be implemented on a principal-by-principal basis. However, a smart card based user will not be able to log in at hosts without smart card readers since there would be no ‘‘password’’ corresponding to the key. This is true of any system which allows the full key space to be used.

If passwords are to co-exist with smart card access (for a given principal) — and, in general, this appears to defeat the purpose of using smart cards for strong authentication — then changes to the registry and authentication server will be required. The registry would have to store both keys, possibly using extended registry attributes (ERAs), and the login script would indicate which of the keys the authentication server should use when encrypting the TGT.

4.1. Smart Card Commands

Memory and communication limitations may preclude loading the entire TGT onto the card at once. Therefore, the TGT may have to be decrypted block by block.

The DCE implementation of the Kerberos protocol uses the encryption algorithm in the cipher block chaining (CBC) mode to encrypt tickets. Since decryption needs to be done block by block, a command to initialise the decryption process is also required. In addition, a command to read the identity of the user from the card is needed. Smart card activation is achieved through yet another command. The precise format of this DCE to card interface requires further study.

4.2. Multiple Card Reader Support

It is undesirable to restrict the brands of cards and card readers which may be used. Unfortunately, the interface from the workstation to the card reader has not been standardised, so separate drivers may be required for every different card reader. By defining an API to the drivers it will be possible for different vendors to provide their own drivers for their card readers. This API will include functions to activate the smart card and deliver ISO format commands to the card.

4.3. Smart Card Login Facility

A replacement login facility will also be required. Such a program will be essentially the same as the DCE `dce_login`, but will make use of the library functions described above to decrypt the TGT on the smart card.

4.4. Management Facilities

The only additional functionality required is a mechanism to synchronise keys on the smart card and in the DCE registry. Kerberos already provides a facility to generate random keys for principals. All that is needed is a routine to program the smart cards using this key. This is an area requiring more development. The system developed by the authors currently requires considerable manual intervention in the setup of the smart cards.

5. ACKNOWLEDGEMENTS

The permission of the Director, Telecom Australia Research Laboratories, to publish this paper is hereby acknowledged.

The authors also wish to acknowledge support and advice from Mark Looi (Queensland University of Technology) and Ed Zuk (Telestra Corporation).

REFERENCES

[BELL] S. Bellovin, M. Merrit, ''Limitations of the Kerberos Authentication System'', Proceedings of Winter USENIX Conference, 1991.

[GONG] L Gong, M. Lomas, R. Needham, J. Saltzer, ''Protecting poorly chosen secrets from guessing attacks'', IEEE Journal on Selected Areas in Communications, Vol 11, No. 5, June 1993.

[ISO] ISO JTC1/SC17/WG4, ''Interindustry Commands for Interchange'', Draft International Standard 7816-4, 1994.

[KIM] G. H. Kim, E. H. Spafford, ''The Design and Implementation of Tripwire: A File System Integrity Checker'', Purdue Technical Report CSD-TR-93-071, November 1993.

[KRAW] M. Krajewski, J. Chipchak, ''Applicability of Smart Cards to Network User Authentication'', Computing Systems, Vol. 7., No. 1, Winter 1994.

[PATO] J. Pato, ''Using Pre-Authentication to Avoid Password Guessing Attacks'', OSF RFC 26.0, June 1993.

AUTHORS' ADDRESSES

Gary Gaskell
Co-operative Research Centre for Distributed Systems Technologies
Queensland University of Technology
Brisbane Queensland 4000
Australia
Telephone: +61-7-864-1051
Fax: +61-7-864-1282
Email: gaskell@dstc.edu.au

Michael Warner
Research Laboratories
Telstra Corporation
770 Blackburn Rd
Clayton Victoria 3168
Australia
Telephone: +61-3-253-6720
Fax: +61-3-253-6352
Email: m.warner@trl.oz.au