Secure RPC Authentication (SRA) for TELNET and FTP

David R. Safford, David K. Hess, and Douglas Lee Schales
Supercomputer Center
Texas A&M University
College Station, TX 77843-3363

1. Abstract

TELNET and FTP currently exchange user authentication (passwords) in plain text, which is easily eavesdropped. Several techniques, such as Kerberos and SPX, have been proposed in draft RFCs to implement secure authentication. These techniques, however, have several drawbacks, including technical complexity, poor vendor support, and organizational problems. This paper presents SRA, a very simple and tested technique based on Secure RPC which, while certainly not as strong as RSA, is reasonably strong, fast, and trivial to implement immediately for both inter and intra-domain communication.

2. Background

TELNET and FTP currently pass the user authentication across the network in the form of plaintext passwords. These passwords can trivially be eavesdropped with such simple tools as etherfind and tcpdump. During intrusions at Texas A&M University in August 1992, significant amounts of the tools used by the crackers were captured. They had much better tools than simple ones such as these, and they used them to capture passwords. It is absolutely essential that TELNET and FTP be extended to provide confidential user authentication to prevent such simple password grabbing.

Several RFCs and draft RFCs address this issue, including:

1409 (TELNET Authentication)
1411 (Kerberos Authentication)
draft-ietf-telnet-authspx-00.txt
draft-ietf-cat-ftpsec-01.txt

These drafts outline proposed designs for the use of Kerberos, RSA, SPX, and gssapi for confidential authentication. While these proposed techniques afford excellent security, they suffer from some drawbacks which have kept them from widespread use so far. Kerberos is difficult to implement and requires centralized ticket servers with copies of all user passwords in plaintext, which is possibly undesirable in large organizations with relatively autonomous subgroups. Inter-domain kerberos is particularly complex. Also, vendors have been extremely slow to provide commercial support for kerberos. Various public key methods can eliminate the need for centralized secret key storage and can solve inter-domain issues very nicely, but need a secure public key distribution system, which will take time to popularize. This paper proposes a simple confidential authentication system for TELNET and FTP based on Secure RPC techniques.

The proposed method has several advantages:

1. The user authentication information is sent encrypted with a one-time DES key.
2. The method uses one-time random Secure RPC keys and, therefore, needs no external keyservers or complex key protection.
3. The method is compatible with all existing and draft authentication methods, so it can take advantage of the method, if supported, transparently on the user's behalf.

4. The method is reasonably fast (it adds about 1 second to a connection on a sparcstation 1 class machine).
5. The method is easily implemented entirely within the client and server binaries, so drop-in installation is very simple. No external key distribution service is needed.
6. All source code is publicly available, making support available for most platforms, independent of vendor support.

One disadvantage of the method is that, unlike more comprehensive public key methods, it does not provide mutual repudiation. The server does validate the user's identity, but the client does not perform server authentication. A second weakness is that Secure RPC is subject to logarithm attacks [1], although such attacks are not trivial. The proposed design for Telnet and FTP reduces this exposure by computing a new random secret/public key pair for each connection. A third possible limitation is that the underlying Diffie-Hellman algorithm is patented. As most vendors already offer Secure RPC as part of Secure NFS implementations, they have already made the necessary key service available.

The bottom line is that the method is significantly more secure than sending passwords in plain text, while offering greater ease of implementation than other, more secure, alternatives.

3. Design

Secure RPC uses a simple version of the Diffie-Hellman exponential based method for determining a common DES key. This algorithm and its specific implementation parameters for Secure RPC are given in the RFC for RPC (RFC 1057). Normally Secure RPC stores a user's key in the keyserver, encrypted with the user’s password, and uses it as necessary to calculate common keys for each connection. In the proposed method, this is greatly simplified by calculating a new random key set for each TELNET or FTP authentication, which eliminates the need for any keyserver. Thus the Secure RPC code is used only for performing basic key calculations, not for key storage or management. The basic data flow is shown in figure 1. The functions performed by the Secure RPC code are indicated by brackets, and the parentheses indicate DES encryption or decryption.

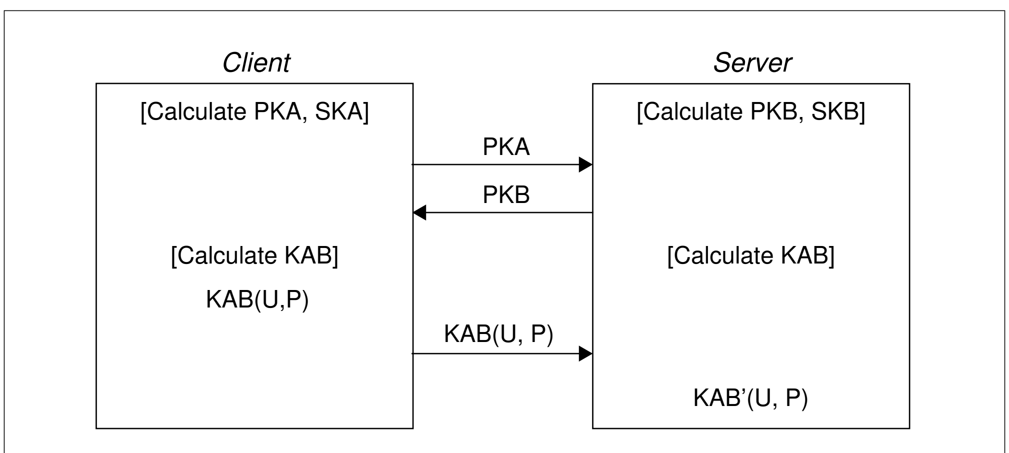

Figure 1. SRA data flow.

From the public Secure RPC code, the keys are 192 bit integers calculated as:

SK (secret key) = random 192 bit number

PK (public key) = (base ^{SK}) mod modulus

where base = 3

modulus = `d4a0ba0250b6fd2ec626e7efd637df76c716e22d0944b88b`

Since exponentiation is commutative, given the key pairs (PKA, SKA) and (PKB, SKB), both sides can confidentially calculate KAB as:

Server: KAB = (PKA ^{SKB}) mod modulus

or

Client: KAB = (PKB ^{SKA}) mod modulus

For purposes of the DES encryption, Secure RPC uses the middle 64 bits of the 192 bit common key (KAB). The calculation of KAB is reasonably secure, as it requires at least one of the secret keys, which are not put on the network, or calculating the logarithm of large numbers.

4. Incorporation into TELNET

The design for incorporation into TELNET followed exactly the design specified by the respective RFC for Kerberos, which is already implemented in the Network Release 2 code. The RFC specifies the necessary option negotiation (WILL, WONT, DO, DONT) to ensure that both sides are able to handle the authentication method. If not, the exchange reverts to the default unencrypted transfer.

SRA has five suboption commands: KEY, USER, PASS, ACCEPT, and REJECT, which are demonstrated in the figure 2.

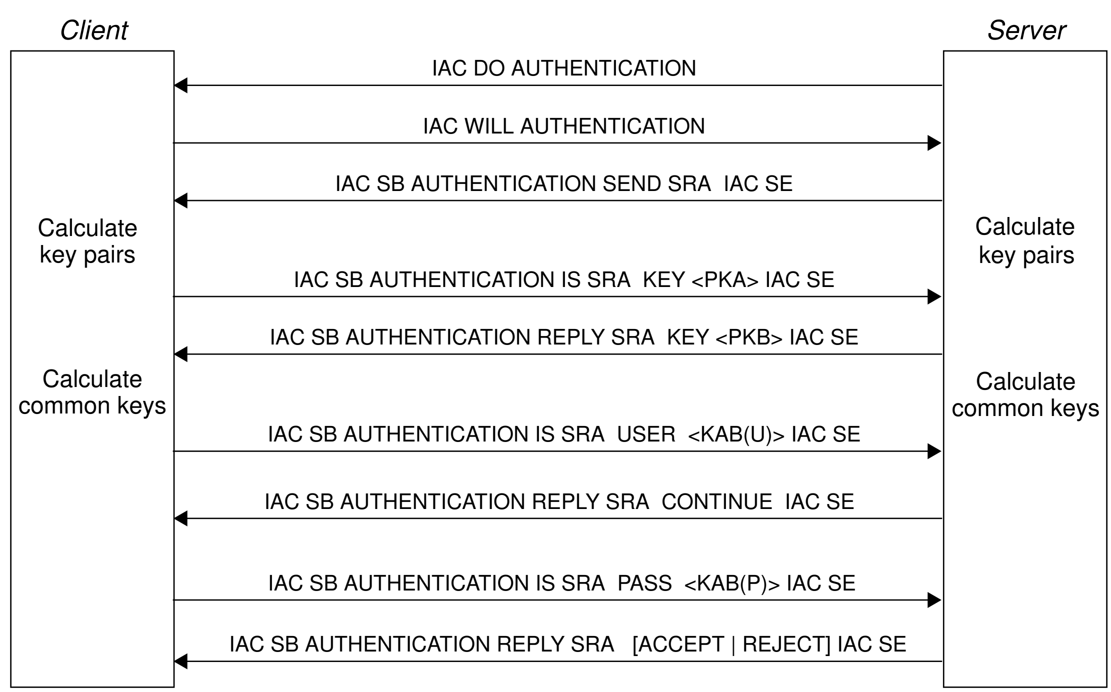

Figure 2. Sample nominal authentication data flow.

5. Incorporation into FTP

The FTP incorporation similarly follows the draft RFC specification for the kerberos exchange (see figure 3), although in this case, the kerberos code was not already implemented in the Network

Release-2.

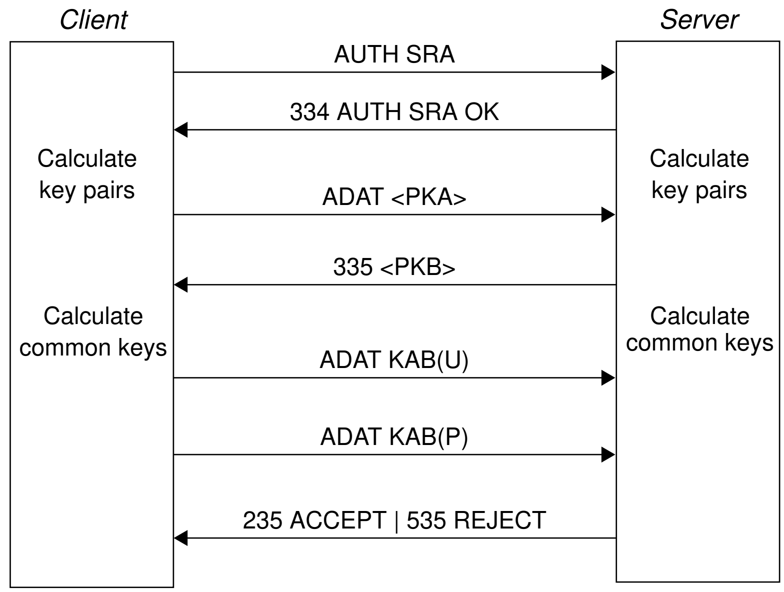

Figure 3. Basic FTP data flow.

6. Preliminary Results

Implementations for both TELNET (telnet and telnetd) and FTP (ftp and ftpd) have been incorporated into the baseline BSD Network Release-2 code and tested on various sparcstation machines running SunOS 4.1.2. The programs have been found to interact well with every existing ftp and telnet service tried, including the major ftp sites, which use experimental servers. Two interesting things have been noticed, which point to further work.

First, this encryption prevents network monitoring programs, such as *Etherscan*, from observing root logins, ftp's and other knob-turning. Overall, this is not too significant a problem, as these events can still be logged by the target service.

A second observation concerns the unnecessary tendency of existing telnet and ftp server code to echo back the user name in plain text following successful authentication, such as in the ftp return message "`230 User smith logged in`". In our implementation we deleted all such direct user identification, although the user's identity will probably still be available through other means, such as rwho and finger. It would be a significant increase in security, regardless of what authentication mechanism is used, to differentiate between a secret login ID and the public user name. The login ID would only be used for authentication purposes (along with the password), while the user name would be used to identify a user publicly. This would be a trivial extension to the shadow password concept, and would be easily implemented on top of the SRA implementation.

7. Availability

The complete source packages for both TELNET and FTP, including all necessary client and server code, (but not including the Secure RPC code), is available on:

sc.tamu.edu:pub/security/SRA.

The relevant Secure RPC code is available in:

ftp.uu.net:networking/rpc/tirpcsrc.tar.Z,

keyserv/setkey.c (routines to calculate common key)

keyserv/generic.c (routines to generate public/secret key pair)

8. References

[1] Brian A. LaMacchia and Andrew M. Odlyzko. "Computation of Discrete Logarithms in Prime Fields", Designs, Codes, and Cryptography, volume 1, 1991, pp46-62.

[2] RFC 1057 "RPC: Remote Procedure Call Protocol Specification, Version 2.", ftp.uu.net:/inet/rfc/rfc1057.Z