

Network Working Group
[draft-brown-pgp-pfs-01.txt](#)
Updates: RFC [2440](#)
Category: INTERNET-DRAFT
Expires: 26 February 2001

I. Brown
University College London
A. Back
Zero-Knowledge Systems
B. Laurie
A.L. Digital Ltd
August 2000

Forward Secrecy Extensions for OpenPGP

Status of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Copyright

Copyright (C) Internet Society 2000. All rights reserved.
Reproduction or translation of the complete document, but not of extracts, including this notice, is freely permitted.

Abstract

The confidentiality of encrypted data depends on the secrecy of the key needed to decrypt it. If one key is able to decrypt large quantities of data, its compromise will be disastrous. This memo describes three methods for limiting this vulnerability for OpenPGP messages: reducing the lifetime of confidentiality keys; one-time keys; and the additional use of lower-layer security services.

Table of Contents

1.	Introduction	2
2.	Short-lifetime encryption keys	3
2.1	Key generation and distribution	3
2.2	Key surrender	4
3.	One-time keys	4
4.	Secure and decentralised e-mail transport	6
5.	Security considerations	6
6.	Acknowledgements	6
7.	Authors	7
8.	References	7
9.	Full Copyright Statement	8

1. Introduction

OpenPGP systems [1] allow two strangers to communicate privately. Each user has a public key that is widely disseminated, and a private key that they keep secret. A message encrypted with a public key can only be decrypted with the related private key. The confidentiality of all messages encrypted with a public key rests on the secrecy of the associated private key.

Online systems such as IPSEC [2] can negotiate new keys for every communication using an algorithm like Diffie-Hellman [3]. If a key is compromised, only the specific session it protected will be revealed to an attacker. This desirable property is called perfect forward secrecy. The security of previous or future encrypted sessions is not affected. Keys are securely deleted after use. Without these keys, there is no way captured ciphertext can be decrypted.

It is more difficult to make store and forward systems like e-mail forward secret, as they rarely make direct connections between a message sender and its recipient. In a typical e-mail encryption system, users create a long-term key pair and publish the public key in a directory, on their Web page, or via other methods. While the use of long-term keys reduces the administrative burden of key distribution, the practice introduces vulnerabilities. If a public key is used for several years, as is common with OpenPGP systems, compromise of the private key will allow an attacker to decrypt any message captured during that time.

This memo describes several methods of reducing the vulnerabilities introduced by use of long-term keys. They are a series of options that MAY be implemented by OpenPGP clients for increased security.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

2. Short-lifetime encryption keys

Using a series of encryption keys, each with a short lifetime, reduces the information revealed by the compromise of any one private key because each key protects less data. If expired keys are securely deleted, attackers will never be able to retrieve them to decrypt captured ciphertext. Therefore when a public encryption key expires, an OpenPGP client **MUST** securely wipe the corresponding private key [4].

Deletion should take place once all messages that could have been sent before expiry have been received and decrypted. For example, as a user logs on, their mail client **SHOULD** retrieve and decrypt all messages from their mail server before deleting any newly-expired private keys. A "panic mode" **MAY** bypass this step.

PGP clients are able to group "subkeys" together under a long-term signature key to signify their common ownership by one principal. To simplify key management, short lifetime keys **SHOULD** be created as subkeys of their owner's long-term signature key.

Clients receiving messages encrypted with an expired key **MAY** warn the sender that they should not use that public key again.

Clients receiving messages encrypted with a revoked key **MUST** warn the sender that they should not use that public key again. Any relevant key revocation certificates **MUST** be included in the warning.

Some OpenPGP systems currently store original message ciphertext and decrypt only for display. While this protects messages on disk, it means that keys must be stored until all messages they protect are deleted. We must assume that an attacker has copies of message ciphertext sent over an insecure network such as the Internet. These messages remain vulnerable until the corresponding private key is deleted.

Messages therefore **MAY** be stored temporarily encrypted with a short-lifetime key, but are unreadable once it has been deleted. Clients **MUST** allow messages to be stored encrypted under a long-term storage key. A mail client **MAY** implement its own secure storage facilities, or use those provided by other software. Messages **SHOULD NOT** be encrypted-to-self using a long-term public key.

2.1 Key generation and distribution

There is a trade-off for the user: the cost of generating and

distributing a new encryption key against the security advantage obtained by earlier key expiry.

Key generation is typically a time-consuming operation. The client SHOULD minimise the time required by the user to complete this operation. This can be achieved, for example, by background key

generation, or by using trade-offs that speed up key generation with minimal reduction in security. With Elgamal [5], for example, the expensive key component to generate is the public prime modulus. A group of keys can share a common public modulus with no negative security implications other than that the key then presents a fatter target for pre-computation attacks. Multiple forward secret Elgamal keys MAY therefore use the same prime modulus with minimal security reduction.

Key distribution can be eased by submitting new keys to key servers, where they will be available for other users to retrieve. Submission and retrieval of generally-available public keys SHOULD be performed automatically by software. Expired public encryption keys MAY be deleted by users and key servers to save space.

If an OpenPGP client has more than one valid encryption key available for a given message recipient, the key nearest its expiration date MUST be used. This limits the time during which the corresponding private key will be vulnerable to attack. The time required to deliver a message should be taken into account by sender and recipient when checking an expiry date.

Signature keys that are long-lived and certified by other users allow a web of trust to build up. Encryption keys SHOULD be certified by a user's long-term signature key to allow their verification by other users.

2.2 Key surrender

Before an OpenPGP client exports a private key as plaintext, the associated public key MUST be revoked and redistributed. A "reason for revocation" signature subpacket MUST be included in the key revocation specifying "Key material has been compromised" (value 0x02).

The least compromising key required MUST be the one surrendered. The session key used to encrypt an individual message will often be sufficient. Otherwise, a subkey should be surrendered before a long-term top-level key. Signature keys should not be surrendered unless absolutely necessary.

3. One-time keys

Taking short-term keys to their logical conclusion, a different key could be used to protect every message. Schneier and Hall [6] suggested a user could make several public keys available in a directory. After a key was retrieved by another user, it would be deleted. This requires message senders to have online access to a

directory. Not all e-mail users have this facility. It also allows an attacker to mount a denial of service attack by exhaustively requesting new one-time keys from the directory.

An off-line scheme is more compatible with the store and forward nature of e-mail, and resistant to DoS. Every time a user sends a

message encrypted with a public key whose signature includes a one-time key support subpacket, they SHOULD include a new one-time public subkey for the recipient to encrypt any reply with. It MUST be sent in the format [primary public signing key | one-time public subkey | signature by primary key]. If PGP/MIME [7] support is available, new key(s) MUST be sent in a separate application/pgp-keys MIME bodypart.

One-time subkeys MUST NOT be exported by their recipient to a third party, particularly a key server.

Users still MUST possess a relatively long-lived encryption key. If Alice were writing to Bob for the first time, she would encrypt her message with his long term key. She would also include a newly created one-time public subkey. Bob would use this new key the next time he wrote to Alice, then delete it. Alice would decrypt the message with the associated private subkey, then securely wipe it.

A "one-time key support" signature subpacket on a public key indicates support for one-time keys. These subpackets MUST be included in the hashed area of a signature. They are formatted as follows:

Subpacket length: 1
Subpacket type: 30

A "one-time key" signature subpacket marker MUST be present in the signature of a one-time subkey. These subpackets are formatted as follows:

Subpacket length: 1
Subpacket type: 158

One-time key subpackets MUST be included in the hashed area of a signature. They are marked as critical so that the entire signature will be ignored by non-compliant OpenPGP clients, preventing more than one message being encrypted using a one-time subkey.

When encrypting messages to a key with a signature containing a one-time key support subpacket, at least one new public encryption subkey MUST be included in the message. This key MUST be signed by the sender's long-term signature key and include a one-time key signature subpacket. The lifetime of a one-time subkey SHOULD be set to as short a period as possible given the expected response time of the recipient. This minimises the key storage requirements of sender and recipient. As a user's collection of private keys grows, she may wish to reduce the lifetime of new one-time subkeys.

A client MUST include further new public encryption subkeys if it believes a message will receive multiple replies. Each reply SHOULD be encrypted with a different subkey if available.

Clients MUST delete a one-time subkey after successfully encrypting data using it. They SHOULD use a one-time subkey, if available, in

preference to a short-lifetime key.

4. Secure and decentralised e-mail transport

The vast majority of current mail clients deliver messages first to a local mail server, which forwards them to their recipient's mail server, where they remain until collected. This procedure minimises security because it fails to take advantage of mail transport protocols such as SMTP [8] over secure transport or network layer security links such as TLS [9] or IPSEC [2]. This is particularly important given that these protocols allow transient keys to be generated and then discarded after each session, providing perfect forward secrecy.

End-to-end security would be better provided if clients delivered messages directly to the recipient's mail server. This allows a secure link to be set up between the two, providing a second layer of forward secrecy. Ideally, as greater numbers of users gain permanent Internet connections through cable modems or Digital Subscriber Lines, they can run mail servers on their own machines. DNS Mail eXchange records [10] can be used to specify a backup mail server such as at an ISP for times when the recipient's machine is unavailable.

OpenPGP mail clients therefore SHOULD deliver messages directly to the recipient's mail server, and SHOULD use any available lower layer security services to protect the links used to deliver messages.

Where OpenPGP keys are used in such services, they SHOULD NOT be used to encrypt keying material that can later be decrypted if they are compromised. Ideally, they SHOULD be used only to authenticate a forward-secret key negotiation protocol such as Diffie-Hellman [3]. At the least, new short-lifetime key pairs SHOULD be generated for key encryption use.

Direct delivery of mail can reveal the sender and recipient of messages to traffic analysts. Clients MAY use anonymous remailers [11] or IP services [12] to mask this information.

5. Security Considerations

Users of these extensions must consider the complete security environment in which they are operating. Highly-secure communications are of limited use between two insecure systems vulnerable to crackers, virii, and other methods of message and key compromise at source. Bellare [13] describes a minimum set of precautions that should be taken.

6. Acknowledgements

Thanks to Nick Bohm, Richard Clayton, Hal Finney and Edwin Woudt for suggestions that have been incorporated into this draft.

7. Authors' Addresses

Ian Brown
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
United Kingdom

Phone: +44 20 7679 3716
Fax: +44 20 7387 1397
E-mail: I.Brown@cs.ucl.ac.uk

Adam Back
Zero-Knowledge Systems Inc.
888 de Maisonneuve East
6th Floor
Montreal
Quebec H2L 4S8
Canada

E-mail: adam@cypherspace.org

Ben Laurie
A.L. Digital Ltd.
Voysey House
Barley Mow Passage
London W4 4GB
United Kingdom

Phone: +44 (20) 8735 0686
E-mail: ben@algroup.co.uk

8. References

- [1] Callas, J., Donnerhackle, L., Finney, H. and Thayer, R.,
"OpenPGP Message Format", [RFC 2440](#), November 1998.
- [2] Atkinson, R., "Security Architecture for the Internet
Protocol", [RFC 1825](#), August 1995.
- [3] Diffie, W. and Hellman, M., "New directions in cryptography",
IEEE Transactions on Information Theory 22(6), November 1976,
644-654.
- [4] US Department of Defense, "Department of Defense Trusted
Computer System Evaluation Criteria", DoD 5200.28-STD,
December 1985.

[5] Elgamal, T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory 31(4), July 1985, 469-472.

[6] Schneier, B. and Hall, C., "An Improved E-mail Security

- Protocol", Proc. 13th Annual Computer Security Applications Conference, New York: ACM Press, 1997, pp. 232-238.
- [7] Elkins, M., Del Torto, D., Levien, R. and Roessler, T., "MIME Security with OpenPGP", IETF work in progress, August 2000.
 - [8] Postel, J., "Simple Mail Transfer Protocol", [RFC 821](#), August 1982.
 - [9] Dierks, T. and Allen, C., "The TLS Protocol", [RFC 2246](#), November 1997.
 - [10] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, [RFC 1034](#), November 1987.
 - [11] Chaum, D., "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM 24(2) 84-88, February 1981.
 - [12] Goldberg, I. and Shostack, A., "Freedom Network 1.0 Architecture and Protocols", <http://www.freedom.net/info/freedompapers/Freedom-Architecture-Protocols.pdf>
 - [13] Bellovin, S., "Can Someone Read My E-Mail?", <http://www.research.att.com/~smb/securemail.html>, 1998.

9. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on

an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
