Network Working Group
Request for Comments: 1991
Category: Informational

D. Atkins
MIT
W. Stallings
Comp-Comm Consulting
P. Zimmermann
Boulder Software Engineering
August 1996

PGP Message Exchange Formats

Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

PGP (Pretty Good Privacy) uses a combination of public-key and conventional encryption to provide security services for electronic mail messages and data files. These services include confidentiality and digital signature. PGP is widely used throughout the global computer community. This document describes the format of "PGP files", i.e., messages that have been encrypted and/or signed with PGP.

PGP was created by Philip Zimmermann and first released, in Version 1.0, in 1991. Subsequent versions have been designed and implemented by an all-volunteer collaborative effort under the design guidance of Philip Zimmermann. PGP and Pretty Good Privacy are trademarks of Philip Zimmermann.

This document describes versions 2.x of PGP. Specifically, versions 2.6 and 2.7 conform to this specification. Version 2.3 conforms to this specification with minor differences.

A new release of PGP, known as PGP 3.0, is anticipated in 1995. To the maximum extent possible, this version will be upwardly compatible with version 2.x. At a minimum, PGP 3.0 will be able to read messages and signatures produced by version 2.x.

2. PGP Services

PGP provides four services related to the format of messages and data files: digital signature, confidentiality, compression, and radix-64 conversion.

2.1 Digital signature

The digital signature service involves the use of a hash code, or message digest, algorithm, and a public-key encryption algorithm. The sequence is as follows:

- -the sender creates a message
- -the sending PGP generates a hash code of the message
- -the sending PGP encrypts the hash code using the sender's private key
- -the encrypted hash code is prepended to the message
- -the receiving PGP decrypts the hash code using the sender's public key
- -the receiving PGP generates a new hash code for the received message and compares it to the decrypted hash code. If the two match, the message is accepted as authentic

Although signatures normally are found attached to the message or file that they sign, this is not always the case: detached signatures are supported. A detached signature may be stored and transmitted separately from the message it signs. This is useful in several contexts. A user may wish to maintain a separate signature log of all messages sent or received. A detached signature of an executable program can detect subsequent virus infection. Finally, detached signatures can be used when more than one party must sign a document, such as a legal contract. Each person's signature is independent and therefore is applied only to the document. Otherwise, signatures would have to be nested, with the second signer signing both the document and the first signature, and so on.

2.2 Confidentiality

PGP provides confidentiality by encrypting messages to be transmitted or data files to be stored locally using conventional encryption. In PGP, each conventional key is used only once. That is, a new key is generated as a random 128-bit number for each message. Since it is to be used only once, the session key is bound to the message and transmitted with it. To protect the key, it is encrypted with the receiver's public key. The sequence is as follows:

- -the sender creates a message
- -the sending PGP generates a random number to be used as a session key for this message only
- -the sending PGP encrypts the message using the session key
- -the session key is encrypted using the recipient's public key and prepended to the encrypted message
- -the receiving PGP decrypts the session key using the recipient's private key

-the receiving PGP decrypts the message using the session key

Both digital signature and confidentiality services may be applied to the same message. First, a signature is generated for the message and prepended to the message. Then, the message plus signature is encrypted using a conventional session key. Finally, the session key is encrypted using public-key encryption and prepended to the encrypted block.

2.3 Compression

As a default, PGP compresses the message after applying the signature but before encryption.

2.4 Radix-64 conversion

When PGP is used, usually part of the block to be transmitted is encrypted. If only the signature service is used, then the message digest is encrypted (with the sender's private key). If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time conventional key). Thus, part or all of the resulting block consists of a stream of arbitrary 8-bit bytes. However, many electronic mail systems only permit the use of blocks consisting of ASCII text. To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters, called ASCII Armor.

The scheme used for this purpose is radix-64 conversion. Each group of three bytes of binary data is mapped into 4 ASCII characters. This format also appends a CRC to detect transmission errors. This radix-64 conversion, also called Ascii Armor, is a wrapper around the binary PGP messages, and is used to protect the binary messages during transmission over non-binary channels, such as Internet Email.

The following table defines the mapping. The characters used are the upper- and lower-case letters, the digits 0 through 9, and the characters + and /. The carriage-return and linefeed characters aren't used in the conversion, nor is the tab or any other character that might be altered by the mail system. The result is a text file that is "immune" to the modifications inflicted by mail systems.

6-bit character value	encoding	6-bit character value	encoding	6-bit character value	encoding	6-bit character value	encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
1	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

It is possible to use PGP to convert any arbitrary file to ASCII Armor. When this is done, PGP tries to compress the data before it is converted to Radix-64.

2.4.1 ASCII Armor Formats

When PGP encodes data into ASCII Armor, it puts specific headers around the data, so PGP can reconstruct the data at a future time. PGP tries to inform the user what kind of data is encoded in the ASCII armor through the use of the headers.

ASCII Armor is created by concatenating the following data:

- An Armor Headerline, appropriate for the type of data
- Armor Headers
- A blank line
- The ASCII-Armored data
- An Armor Checksum
- The Armor Tail (which depends on the Armor Headerline).

An Armor Headerline is composed by taking the appropriate headerline text surrounded by five (5) dashes (-) on either side of the headerline text. The headerline text is chosen based upon the type of data that is being encoded in Armor, and how it is being encoded. Headerline texts include the following strings:

BEGIN PGP MESSAGE -- used for signed, encrypted, or compressed files
BEGIN PGP PUBLIC KEY BLOCK -- used for transferring public keys

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.