Internet Engineering Task Force
Internet-Draft
Updates: 4880 (if approved)
Intended status: Standards Track
Expires: December 11, 2015

D. Atkins
SecureRF Corporation

June 09, 2015

OpenPGP Extensions for Device Certificates
draft-atkins-openpgp-device-certificates-03

Abstract

The OpenPGP Message Formats defined in RFC 4880 specify packet formats and methods for combining those packets to form messages and certificates. However RFC 4880 made an architectural decision that keys are owned by users and must be self-certified. New use cases have emerged where that is not the case. There is a desire to have certificates that are not tied to a user (e.g. device certificates) which may only have encryption keys so may not be self certifiable. Moreover, devices might be space constrained so reducing size is important. This draft specifies extensions to (and updates) RFC 4880 that loosen the definitions of certificates in order to enable userless certificates without self-certifications and specifies a set of notations to enable compact device certifications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on December 11, 2015.

Table of Contents

1. Introduction

OpenPGP [RFC4880] defines a standard, compact format for, among other things, sharing keys and signature certificates. Unfortunately the specification is user-focused, assuming that there are people sitting at the ends and creating and managing those keys. New use cases have come up where that is not the case and the endpoint for these keys are devices, not users. Yet we still want to be able to certify these device keys.

Since the publication of RFC 4880, new use cases have emerged that don't fit into the existing standard models. For example, the

Internet of Things have introduced devices that need to certify device-level encryption keys but cannot self-certify and have no user associated. This draft suggests extensions to RFC 4880 that enable those use cases and make it easier to have device certificates using OpenPGP.

1.1. Requirements Language

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 [RFC2119].

2. Device Certificate Format

RFC 4880 section 12.1 defines a v4 Public Key Format as a sequence of packets starting with a Primary Key and then a sequence of packets and subpackets that add Revocations, User IDs, and Signatures.

The description in RFC 4880 requires a User ID. Implementors of this specification can loosen that requirement such that an augmented V4 device certificate looks like the following sequence (no longer requiring a User ID packet):

```
Primary-Key
   [Revocation Self Signature]
   [Direct Key Signature...]
   [User ID [Signature ...] ...]
   [User Attribute [Signature ...] ...]
   [[Subkey [Binding-Signature-Revocation]
           Primary-Key-Binding-Signature] ...]
```

Note that RFC 4880 section 11.1 defines this same sequence in text for transferable public keys. Implementors of this specification can change that definition from "One or more User ID packets" to "Zero or more User ID packets".

Moreover, one more relaxation from section 12.1. RFC 4880 states that "In a V4 key, the primary key MUST be a key capable of certification." Implementors of this specification can loosen that restriction as well, such that in V4 augmented key, the primary key MAY be a key capable of certification.

A primary key capable of making signatures SHOULD be accompanied by either a certification signature (on a User ID or User Attribute) or a signature directly on the key.

Implementations MUST accept encryption-only primary keys without a signature. It also MUST allow importing any key accompanied either by a certification signature or a signature on itself. It MAY accept signature-capable primary keys without an accompanying signature.

3. User ID Attribute Subpacket

Section 5.12 of RFC 4880 defines the User Attribute Packet which can be used in lieu of a User ID Packet. Whereas the User ID Packet only allows a single UTF-8 string content, the User Attribute Packet allows the addition of multiple attributes in subtype packets. Unfortunately RFC 4880 only defined a single Attribute Subpacket, the Image Attribute. This means that you need two signatures if you want to have an ID and an image.

To solve that problem for device certificates we define a new User Attribute Subpacket, the User ID Attribute Subpacket, type #[IANA -- assignment TBD1]. A User ID Attribute subpacket, just like a User ID packet, consists of UTF-8 text that is intended to represent the name and email address of the key holder. By convention, it includes an RFC 2822 [RFC2822] mail name-addr, but there are no restrictions on its content. For devices, it may be the device identifier. The packet length in the header specifies the length of the User ID.

Note that RFC 4880 already allows a User Attribute packet anywhere a User ID packet can be used. See RFC 4880 section 5.2.3.19 (Primary User ID) for more information on self-signatures over these kinds of packets. Any signature on a User Attribute packet covers all subpackets. Implementations MAY decide to trust the User ID Subpacket.

4. Device Certification Notations

OpenPGP defines a signature notation data packet that allows implementors and users to add extra data to signatures. RFC 4880 defined a registry for a global namespace and requires using name@dom.ain domain-name notations otherwise. Many of the devices targeted by this specification have limited storage capability, so it behooves an implementor to limit the extraneous storage.

These notations can be important when you have a third-party device certification. That third party might want to add extra data about the device to its signature certification. In order to keep the certificate smaller we define a set of notations that MAY be used when signing a device certificate.

4.1. The 'manu' Notation

The "manu" notation is a string that declares the device manufacturer's name. The certifier key is asserting this string (which may or may not be related to the User ID of the certifier's key).

4.2. The 'make' Notation

This notation defines the product make. It is a free form string.

4.3. The 'model' Notation

This notation defines the product model name/number. It is a free form string.

4.4. The 'prodid' Notation

This notation contains the product identifier. It is a free form string.

4.5. The 'pvers' Notation

This notation defines the product version number (which could be a release number, year, or some other identifier to differentiate different versions of the same make/model). It is a free form string.

4.6. The 'lot' Notation

This notation defines the product lot number (which is an indicator of the batch of product). It is a free form string.

4.7. The 'qty' Notation

This notation defines the quantity of items in this package. It is a decimal integer representation with no punctuation, e.g. "10", "1000", "10000", etc.

4.8. The 'loc' and 'dest' Notations

The "loc" and 'dest' notations declare a GeoLocation as defined by RFC 5870 [RFC5870] but without the leading "geo:" header. For example, if you had a GeoLocation URI of "geo:13.4125,103.8667" you would encode that in these notations as "13.4125,103.8667".

The 'loc' notation is meant to encode the geo location where the signature was made. The 'dest' notation is meant to encode the geo location where the device is "destined" (i.e., a "destination" for the device).

4.9. The 'hash' Notation

A 'hash' notation is a means to include external data in the contents of a signature without including the data itself. This is done by hashing the external data separately and then including the data's name and hash in the signature via this notation. This is useful, for example, to have an external "manifest," "image," or other data that might not be vital to the signature itself but still needs to be protected and authenticated without requiring a second signature.

The 'hash' notation has the following structure:

- A single byte specifying the length of the name of the hashed data
- A UTF-8 string of the name of the hashed data
- A single byte specifying the hash algorithm (see RFC 4880 section 9.4)
- The binary hash output of the hashed data using the specified algorithm. (The length of this data is implicit based on the algorithm specified).

Due to its nature a 'hash' notation is not human readable and MUST NOT be marked as such when used.

5. Acknowledgements

A big thank you to Werner Koch, David Shaw, Jon Callas, Daniel Nagy, and David Leon Gil for their input on the concepts and text of this document.

6. IANA Considerations

This document requests IANA to register several items within the OpenPGP parameters (or the "name space" in the terminology of [RFC5226]) created by [RFC4880].

6.1. PGP User Attribute Types

This specification asks IANA to register a PGP User Attribute Type:

Value	Attribute	Reference
TBD1	User ID	This Doc Section 3

Table 1: User Attribute Types

6.2. Signature Notation Data Subpacket Types

This specification asks IANA to register a set of OpenPGP Signature Notation Data Subpacket Types defined in Section 4. The following table is a summary of the requested registrations.

Allowed Values	Name	Type	Reference
Any String	manu	Manufacturer Name	This doc Section 4.1
Any String	make	Product Make	This doc Section 4.2
Any String	model	Product Model	This doc Section 4.3
Any String	prodid	Product ID	This doc Section 4.4
Any String	pvers	Product Version	This doc Section 4.5
Any String	lot	Product Lot Number	This doc Section 4.6
Decimal Integer String	qty	Package Quantity	This doc Section 4.7
A geo: URI without the "geo:"	loc	Current Geo-location Latitude/Longitude	This doc Section 4.8
A geo: URI without the "geo:"	dest	Destination Geo-location Latitude/Longitude	This doc Section 4.8

Hash Notation data	hash	The Hash of external data	This doc Section 4.9

Table 2: Device Certificate Notations

7. Security Considerations

The Security Considerations of [RFC4880] apply.

OpenPGP was designed with security in mind, with many smart, intelligent people spending a lot of time thinking about the ramifications of their decisions. Removing the requirement for self-certifying User ID (and User Attribute) packets on a key means that someone could surreptitiously add an unwanted ID to a key and sign it. If enough "trusted" people sign that surreptitious identity then other people might believe it. The attack could wind up sending encrypted mail destined for alice to some other target, bob, because someone added "alice" to bob's key without bob's consent.

In the case of device certificates the device itself does not have any consent. It is given an identity by the device manufacturer and the manufacturer can insert that ID on the device certificate, signing it with the manufacturer's key. If another people wants to label the device by another name, they can do so. There is no harm in multiple IDs, because the verification is all done based on who has signed those IDs.

When a key can self-sign, it is still suggested to self-certify IDs, even if it no longer required by this modification to OpenPGP. This at least signals to recipients of keys that yes, the owner of this key asserts that this identity belongs to herself. Note, however, that mallet could still assert that he is 'alice' and could even self-certify that. So the attack is not truly different. Moreover, in the case of device certificates, it's more the manufacturer than the device that wants to assert an identity (even if the device could self-certify).

There is no signaling whether a key is using this new, looser-requirement key format. An attacker could therefore just remove the self-signature off a published key. However one would hope that wide publication would result in another copy still having that signature and it being returned quickly. However, the lack of signaling also means that a user with an application following RFC 4880 directly would see a key following this specification as "broken" and may not accept it.

On a different note, including the "geo" notation could leak information about where a signer is located. However it is just an assertion (albeit a signed assertion) so there is no verifiable truth to the location information released. Similarly, all the rest of the signature notations are pure assertions, so they should be taken with the trustworthiness of the signer.

Combining the User ID with the User Attribute means that an ID and image would not be separable. For a person this is probably not good, but for a device it's unlikely the image will change so it makes sense to combine the ID and image into a single signed packet with a single signature.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

8.2. Informative References

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC5870] Mayrhofer, A. and C. Spanring, "A Uniform Resource Identifier for Geographic Locations ('geo' URI)", RFC 5870, June 2010.

Author's Address

Derek Atkins
SecureRF Corporation
100 Beard Sawmill Rd, Suite 350
Shelton, CT 06484
US

Phone: +1 617 623 3745
Email: datkins@securerf.com, derek@ihtfp.com