

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: November 19, 2018

W. Koch  
GnuPG e.V.  
May 18, 2018

**OpenPGP Web Key Directory**  
**draft-koch-openpgp-webkey-service-06**

**Abstract**

This specification describes a service to locate OpenPGP keys by mail address using a Web service and the HTTPS protocol. It also provides a method for secure communication between the key owner and the mail provider to publish and revoke the public key.

**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 November 19, 2018.

**Copyright Notice**

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Notational Conventions . . . . .	<a href="#">2</a>
<a href="#">3.</a>	Web Key Directory . . . . .	<a href="#">2</a>
<a href="#">3.1.</a>	Key Discovery . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Web Key Directory Update Protocol . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	The Submission Address . . . . .	<a href="#">6</a>
<a href="#">4.2.</a>	The Submission Mail . . . . .	<a href="#">6</a>
<a href="#">4.3.</a>	The Confirmation Request . . . . .	<a href="#">7</a>
<a href="#">4.4.</a>	The Confirmation Response . . . . .	<a href="#">8</a>
<a href="#">4.5.</a>	Policy Flags . . . . .	<a href="#">9</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">10</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">10</a>
<a href="#">6.1.</a>	Well-Known URI . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Acknowledgments . . . . .	<a href="#">11</a>
<a href="#">8.</a>	Normative References . . . . .	<a href="#">11</a>
<a href="#">Appendix A.</a>	Sample Protocol Run . . . . .	<a href="#">11</a>
<a href="#">A.1.</a>	Sample Keys . . . . .	<a href="#">12</a>
<a href="#">A.2.</a>	Sample Messages . . . . .	<a href="#">13</a>
<a href="#">Appendix B.</a>	Changes Since -05 . . . . .	<a href="#">16</a>
Author's Address	. . . . .	<a href="#">17</a>

## [1.](#) Introduction

This memo describes a method to associate OpenPGP keys with a mail address and how to look them up using a web service with a well-known URI. In addition a mail based protocol is given to allow a client to setup such an association and to maintain it.

## [2.](#) Notational Conventions

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 [[RFC2119](#)].

## [3.](#) Web Key Directory

A major use case for OpenPGP is the encryption of mail. A common difficulty of sending encrypted mails to a new communication partner is to find the appropriate public key of the recipient. Unless an off-channel key exchange has been done, there are no easy ways to discover the required key. The common practice is to search the network of public key servers for a key matching the recipient's mail address. This practise bears the problem that the key servers are not able to give a positive confirmation that a key actually belongs to the mail addresses given in the key. Further, there are often several keys matching a mail address and thus one needs to pick a key



on good luck. This is clearly not a secure way to setup an end-to-end encryption. Even if the need for a trusted key for an initial mail message is relinquished, a non-authenticated key may be a wrong one and the actual recipient would receive a mail which she can't decrypt, due to the use of a wrong key.

Methods to overcome this problem are

- o sending an initial unencrypted message with the public key attached,
- o using the OpenPGP DANE protocol to lookup the recipients key via the DNS.

The first method has the obvious problems of not even trying to encrypt the initial mail, an extra mail round-trip, and problems with unattended key discovery.

The latter method works fine but requires that mail providers need to set up a separate DNS resolver to provide the key. The administration of a DNS zone is often not in the hands of small mail installations. Thus an update of the DNS resource records needs to be delegated to the ISP running the DNS service. Further, DNS lookups are not encrypted and missing all confidentiality. Even if the participating MUAs are using STARTTLS to encrypt the mail exchange, a DNS lookup for the key unnecessarily identifies the local-part of the recipients mail address to any passive eavesdroppers.

This memo specified a new method for key discovery using an encrypted https connection.

### **3.1. Key Discovery**

Although URIs are able to encode all kind of characters, straightforward implementations of a key directory may want to store the local-part of a mail address directly in the file system. This forbids the use of certain characters in the local-part. To allow for such an implementation method the URI uses an encoded form of the local-part which can be directly mapped to a file name.

OpenPGP defines its User IDs, and thus the mail address, as UTF-8 strings. To help with the common pattern of using capitalized names (e.g. "Joe.Doe@example.org") for mail addresses, and under the premise that almost all MTAs treat the local-part case-insensitive and that the domain-part is required to be compared case-insensitive anyway, all upper-case ASCII characters in a User ID are mapped to lowercase. Non-ASCII characters are not changed.



The so mapped local-part is hashed using the SHA-1 algorithm. The resulting 160 bit digest is encoded using the Z-Base-32 method as described in [\[RFC6189\]](#), [section 5.1.6](#). The resulting string has a fixed length of 32 octets. To form the URI, the following parts are concatenated:

- o The scheme "https://",
- o the domain-part,
- o the string "/.well-known/openpgpkey/hu/",
- o and the above constructed 32 octet string.

For example the URI to lookup the key for Joe.Doe@Example.ORG is:

```
https://example.org/.well-known/openpgpkey/  
hu/iy9q119eutrkn8s1mk4r39qejnbu3n5q
```

(line has been wrapped for rendering purposes)

DNS SRV resource records ([\[RFC2782\]](#)) may be used to query a different host or a port other than 443. For example:

```
_openpgpkey._tcp.example.org. IN SRV 0 0 8443 wkd.example.org.
```

changes the above to query the host "wkd.example.org" at port 8443 instead of the host "example.org" at port 443. The target (in the example "wkd.example.org") MUST be a sub-domain of the domain-part (here "example.org"). If the target is not a sub-domain, the SRV RR MUST be ignored. The recommended name for the sub-domain is "wkd".

The HTTP GET method MUST return the binary representation of the OpenPGP key for the given mail address. The key needs to carry a User ID packet ([\[RFC4880\]](#)) with that mail address. Note that the key may be revoked or expired - it is up to the client to handle such conditions. To ease distribution of revoked keys, a server may return revoked keys in addition to a new key. The keys are returned by a single request as concatenated key blocks.

The server MUST accept the HTTP HEAD method to allow a client to check for the existence of a key.

The server SHOULD use "application/octet-string" as the Content-Type for the data but clients SHOULD also accept any other Content-Type. The server MUST NOT return an ASCII armored version of the key.



The server MUST serve a Policy Flags file as specified below. That file is even required if the Web Key Directory Update Protocol is not supported.

#### **4. Web Key Directory Update Protocol**

To put keys into the key directory a protocol to automate the task is desirable. The protocol defined here is entirely based on mail and the assumption that a mail provider can securely deliver mail to the INBOX of a user (e.g. an IMAP folder). Note that the same protocol may also be used for submitting keys for use with OpenPGP DANE.

We assume that the user already created a key for her mail account `alice@example.org`. To install the key at her provider's Web Key Directory, she performs the following steps:

1. She retrieves a file which contains one line with the mail address used to submit the key to the mail provider. The DNS SRV rules described for the Web Key Directory apply here as well. See below for the syntax of that file. For a mail address at the domain "example.org" the URI of the file is

`https://example.org/.well-known/openpgpkey/submission-address`

2. She sends her key using SMTP (or any other transport mechanism) to the provider using the submission address and key format as specified by PGP/MIME.
3. The provider checks that the received key has a User ID which matches an account name of the provider.
4. The provider sends an encrypted message containing a nonce and the fingerprint of the key to the mail account of the user. Note that a similar scheme is used by the well known `caff(1)` tool to help with key signing parties.
5. A legitimate user will be able to decrypt the message because she created the key and is in charge of the private key. This step verifies that the submitted key has actually been created by the owner of the account.
6. The user sends the decrypted nonce back to the submission address as a confirmation that the private key is owned by her and that the provider may now publish the key. Although technically not required, it is suggested that the mail to the provider is encrypted. The public key for this is retrieved using the key lookup protocol described above.





7. The provider receives the nonce, matches it with its database of pending confirmations and then publishes the key. Finally the provider sends a mail back to the user to notify her of the publication of her key.

The message data structures used for the above protocol are specified in detail below. In the following sections the string "WELLKNOWN" denotes the first part of an URI specific for a domain. In the examples the domain "example.org" is assumed, thus

WELLKNOWN := `https://example.org/.well-known/openpgpkey`

The term "target key" denotes the to be published key, the term "submission key" the key associated with the submission-address of the mail provider.

#### **4.1. The Submission Address**

The address of the submission file is

WELLKNOWN/submission-address

The file consists of exactly one line, terminated by a LF, or the sequence of CR and LF, with the full mail address to be used for submission of a key to the mail provider. For example the content of the file may be

key-submission-example.org@directory.example.org

#### **4.2. The Submission Mail**

The mail used to submit a key to the mail provider MUST comply to the PGP/MIME specification ([\[RFC3156\]](#), [section 7](#)), which states that the Content-Type must be "application/pgp-keys", there are no required or optional parameters, and the body part contains the ASCII-armored transferable Public Key Packets as defined in [\[RFC4880\]](#), [section 11.1](#).

The mail provider MUST publish a key capable of signing and encryption for the submission-address in the Web Key Directory or via DANE. The key to be published MUST be submitted using a PGP/MIME encrypted message ([\[RFC3156\]](#), [section 4](#)). The message MUST NOT be signed (because the authenticity of the signing key has not yet been confirmed). After decryption of the message at the mail provider a single "application/pgp-keys" part, as specified above, is expected.



### **4.3. The Confirmation Request**

The mail provider sends a confirmation mail in response to a received key publication request. The message **MUST** be sent from the submission-address of the mail provider to the mail address extracted from the target key. The message needs to be a PGP/MIME signed message using the submission key of the provider for the signature. The signed message **MUST** have two parts:

The first part **MUST** have "text" as its Content-Type and can be used to explain the purpose of the mail. For example it may point to this RFC and explain on how to manually perform the protocol.

The second part **MUST** have "application/vnd.gnupg.wkd" if the protocol version of the server is 5 or later; without a known protocol version or a version less than 5, "application/vnd.gnupg.wks" **MUST** be used as its Content-Type and carry an OpenPGP encrypted message in ASCII Armor format. The message **MUST** be encrypted to the target key and **MUST NOT** be signed. After decryption a text file in the Web Key data format must be yielded.

That data format consists of name-value pairs with one name-value pair per LF or CR+LF terminated line. Empty lines are allowed and will be ignored by the receiver. A colon is used to terminate a name.

In a confirmation request the following names **MUST** be send in the specified order:

- o "type": The value must be "confirmation-request".
- o "sender": This is the mailbox the user is expected to sent the confirmation response to. The value must match the mailbox part of the "From:" address of this request. Exactly one address **MUST** be given.
- o "address": The value is the addr-spec part of the target key's mail address. The value **SHOULD** match the addr-spec part of the recipient's address. The value **MUST** be UTF-8 encoded as required for an OpenPGP User ID.
- o "fingerprint": The value is the fingerprint of the target key. The fingerprint is given in uppercase hex encoding without any interleaving spaces.
- o "nonce": The value is a string with a minimum length of 16 octets and a maximum length of 64 octets. The string must entirely be made up of random ASCII letters or digits. This nonce will be



sent back to the mail provider as proof that the recipient is the legitimate owner of the target-key.

The receiver of that message is expected to verify the outer signature and disregard the entire message if it can't be verified or has not been signed by the key associated with the submission address.

After the message has been verified the receiver decrypts the second part of the message, checks that the "fingerprint" matches the target key, checks that the "address" matches a User ID of the target key, and checks the other constraints of the request format. If any constraint is not asserted, or the fingerprint or User ID do not match the target key, or there is no pending publication requests (i.e. a mail recently sent to the submission address), the user MAY be notified about this fake confirmation attempt.

In other cases the confirmation request is legitimate and the MUA shall silently send a response as described in the next section.

The rationale for the outer signature used with this request is to allow early detection of spam mails. This can be done prior to the decryption step and avoids asking the user to enter a passphrase to perform the decryption for a non-legitimate message. The use of a simple encrypted attachment, instead of using PGP/MIME encryption, is to convey the Content-Type of that attachment in the clear and also to prevent automatic decryption of that attachment by PGP/MIME aware clients. The MUA may in fact detect this confirmation request and present a customized dialog for confirming that request.

#### **4.4. The Confirmation Response**

A response to a confirmation request MUST only be sent in the positive case; there is no negative confirmation response. A mail service provider is expected to cancel a pending key submission after a suitable time without a confirmation. The mail service provider SHOULD NOT retry the sending of a confirmation request after the first request has been sent successfully.

The user MUST send the confirmation response from her target mail address to the "from" address of the confirmation request. The message MUST be signed and encrypted using the PGP/MIME Combined format ([\[RFC3156\]](#), [section 6.2](#)). The signing key is the target key and the encryption key is the key associated with the provider's submission address.

The Content-Type used for the plaintext message MUST match the Content-Type of the request. The format is the same as described



above for the Confirmation Request. The body must contain three name-value pairs in this order:

- o "type": The value must be "confirmation-response".
- o "sender": The value must match the mailbox part of the "From:" address of this response. Exactly one address MUST be given.
- o "nonce": The value is the value of the "nonce" parameter from the confirmation request.

#### **4.5. Policy Flags**

For key generation and submission it is useful to tell the client about certain properties of the mail provider in advance. This can be done with a file at the URL

WELLKNOWN/policy

A site supporting the Web Key Directory MUST serve this file; it is sufficient if that file has a zero length. Clients may use this file to check for Web Key Directory support.

The file contains keywords and optionally values, one per line with each line terminated by a LF or the sequence of CR and LF. Empty lines and lines starting with a '#' character are considered comment lines. A keyword is made up of lowercase letters, digits, hyphens, or dots. An underscore is allowed as a name space delimiters; see below. The first character must be a letter. Keywords which are defined to require a value are directly followed by a colon and then after optional white space the value. Clients MUST use case-insensitive matching for the keyword.

Currently defined keywords are:

- o "mailbox-only": The mail server provider does only accept keys with only a mailbox in the User ID. In particular User IDs with a real name in addition to the mailbox will be rejected as invalid.
- o "dane-only": The mail server provider does not run a Web Key Directory but only an OpenPGP DANE service. The Web Key Directory Update protocol is used to update the keys for the DANE service.
- o "auth-submit": The submission of the mail to the server is done using an authenticated connection. Thus the submitted key will be published immediately without any confirmation request.





- o "protocol-version": This keyword can be used to explicitly claim the support of a specific version of the Web Key Directory update protocol. This is in general not needed but implementations may have workarounds for providers which only support an old protocol version. If these providers update to a newer version they should add this keyword so that the implementation can disable the workaround. The value is an integer corresponding to the respective draft revision number.
- o "submission-address": An alternative way to specify the submission address. The value is the addr-spec part of the address to send requests to this server. If this keyword is used in addition to the "submission-address" file, both MUST have the same value.

More keywords will be defined in updates to this I-D. There is no registry except for this document. For experimental use of new features or for provider specific settings, keywords MUST be prefixed with a domain name and an underscore.

## **5. Security Considerations**

The use of SHA-1 for the mapping of the local-part to a fixed string is not a security feature but merely used to map the local-part to a fixed-sized string made from a well defined set of characters. It is not intended to conceal information about a mail address.

The domain name part of the mail address is not part of the hash to avoid problems with internationalized domain names. Instead a separate URL is required for each domain name.

The use of DNS SRV records reduces the certainty that a mail address belongs to a domain. For example an attacker may change the target to a host in a sub-domain under their control and thus gain full control over all keys. An implementation may want to weight the certainty of a mapping different if it has been retrieved via a sub-domain and in particular if a non-recommended name is used for the sub-domain.

To make it a bit harder to test for published keys, the server responsible to serve the WELLKNOWN directory SHOULD NOT create an index file for that directory or any sub-directory.

## **6. IANA Considerations**



### **6.1. Well-Known URI**

IANA is requested to assign a well-known URI in the "Well-Known URIs" registry as defined by [[RFC5785](#)]:

URI suffix: openpgpkey

Change controller: IETF

Specification document: This

## **7. Acknowledgments**

The author would like to acknowledge the help of the individuals who kindly voiced their opinions on the GnuPG mailing lists, in particular, the help of Bernhard Reiter and Guilhem Moulin.

## **8. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), DOI 10.17487/RFC2782, February 2000, <<https://www.rfc-editor.org/info/rfc2782>>.
- [RFC3156] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", [RFC 3156](#), August 2001.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", [RFC 4880](#), November 2007.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), DOI 10.17487/RFC5785, April 2010, <<http://www.rfc-editor.org/info/rfc5785>>.
- [RFC6189] Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", [RFC 6189](#), DOI 10.17487/RFC6189, April 2011, <<http://www.rfc-editor.org/info/rfc6189>>.

## **Appendix A. Sample Protocol Run**

The following non-normative example can be used by implementors as guidance.



Note that GnuPG version 2.1.12 supports the key discovery described in version -00 of this document (auto-key-locate method "wkd"). Version 2.1.16 can run the protocol described in this document but is also able to run the protocol version specified by -01. For backward compatibility this example uses the Content-Type as required for versions of this protocol prior to -04; if the client knows that the server support -04 "vnd.gnupg.wkd" should be used.

### [A.1.](#) Sample Keys

This is the provider's submission key:

-----BEGIN PGP PRIVATE KEY BLOCK-----

```
lFgEV/TAohYJKwYBBAHaRw8BAQdAB/k9YQfSTI8qQqqK1KimH/BsvzsowWItSQPT
FP+fOC4AAP46uJ3Snno3Vy+kORye3rf0VvWvuz82voEQLxG6WpfHhREEtBprZXkt
c3VibWlzc2lrbkBlcGFtcGxlM5ldIh5BBMWCAAhBQJX9MCiAhsDBQsJCACCBhUI
CQoLAgQWAgMBAh4BAheAAAOJEKhtNoow0cqEWMUA/0e9XaeptszWC9ZvPg8INL6a
BvRqPBYGU7PGmuXsxBoVADyck0ykG0UAFHVyN1w4gSK/biMcqVr857i8/HuvjW
C5xdBFf0wKISCisGAQQBl1UBBQEBB0Apva0e4MtSEJ1fpds/4DF12kXXBpnVji/s
Wg9btdthNQMBACAAAP9FJX99T1LEJzBnvBBnc6bimnT6/10KM9Rd04R0/uVP6BFL
iGEEGBYIAAKFAIf0wKICGwACGkQqG02ihbRyoTlGwD9FBr92osjL7HkhhZZ7Z2D
My3b9zpoZeMjvPg5YPqpdKMA/jhZoHuZCRMBYf7YRFb8aXtuyetDFZYrkjnum+OG
HFAD
=Hnwd
-----END PGP PRIVATE KEY BLOCK-----
```

This is the target key to be published:

-----BEGIN PGP PRIVATE KEY BLOCK-----

```
lFgEV2o9XRYJKwYBBAHaRw8BAQdAZ8zkuQDL9x7rcvvoo6s3iEF1j88Dknd9nZhL
nTEoBRkAAP94nCZMM4WY2IORXfM6phLGSz3RsHvs/vA10paus4+R3BKJtBtwYXRy
aWNlLmx1bXVtYmFAZXhhbXBsZS5uZXSIeQQTfGgAIQUcV2o9XQIbAwULCQgHAgYV
CAkKCwIEFgIDAQIeAQIXgAAKCRATlWNoKgINCpkNAQDFDcwJUzsxu7aJUiPdpYXj
4uVarrXakxEE8mGFotWhLAD9GH4rqLDYIE3NKEU0s+0kt4tEIwJaV8H1NNPPPMiK
3g2cXQRXaj2NEgorBgEEAZdVAQUBAQdAFnmZc99TuKk5iCq9wmYZUVF2RcXN2Cs
qAl8iGQQUwsDAQgHAAD/VN/VGmlcwGBPcLTya2hfU4t37nMcFCKdNSXjJ5DFA0AP
PohhBBgWCAAJBQJXaj2NAhsMAAOJEBOVY2gqAg0Ky4UA/0GmVaXzXemLvv1Xw4yx
Eaz/KfKKGc4RJ+38fyqUzw8NAQCohQ+ki3I5f84EXLZEiUiLsnVtOn1HNxvND/gw
TiFZBA==
=GHi7
-----END PGP PRIVATE KEY BLOCK-----
```



## [A.2.](#) Sample Messages

The first message triggers the publication requests.

```
From: patrice.lumumba@example.net
To: key-submission@example.net
Subject: Key publishing request
MIME-Version: 1.0
Content-Type: multipart/encrypted;
              protocol="application/pgp-encrypted";
              boundary="=-=01-e8k41e11ob31eefa36wo=-="
Date: Wed, 05 Oct 2016 10:15:51 +0000
```

```
--=-=01-e8k41e11ob31eefa36wo=-=
Content-Type: application/pgp-encrypted
```

```
Version: 1
```

```
--=-=01-e8k41e11ob31eefa36wo=-=
Content-Type: application/octet-stream
```

```
-----BEGIN PGP MESSAGE-----
```

```
hF4DUgLY5tvmW2sSAQdAR1AcqvFpQe/fHRZbf0xcn19Tb+AtwaX2yZnZXGELGHsw
1/e3E0JptwM5tpRAVe71ooF8Zq4j176ZgQKfj/SyjpLJxyoEDy2N5wTQaqW4JtML
0ukB1vh7dIRDxBJX/LQIJC0wz8o1Q3vjclJKFFvDb7YrerABpPIzw0AupcgIbQHj
5m1+2WU5CL8ffYJy2h1jV2X40nvWF1Sn6J6SVD6DfZp0PRt9TxSemJrN1LJ3lG0N
ts8AuYmC0eC1H2r5TYyxqkC98JF8+Nvyxd/fwne8IOjk9uixkNMC5H9/ZOH0YWcb
wBnNB4iXuym40IPxiLkDymsVF0ww/XrODE9Y259EGm045VFNrJAX3HFs9/PcMCVkn
2qMyEkr8LHiXeEPun6Z54RHUPYv2cUkEZ0hhSJ+rtBxkc/5D/cAScCEXRKFSKEF
jLJAvLK/u/ga5DAzVai+vh6b6Bq+YVPaD9GWMhwj4CgR90p9LULi6S/Hzwhv9Wzf
8fJoJ0aDjyvRDgr09jYLWamxkS9NWxqwy6MXJvxbNdd5XtqiW4Y4o0L1hDJhXR
lJn/XvotXKwhKN+4QghIXDvt4Dl4XxS5ptwVTau8W8DYqDsU2obEcfsirZv53M1
Q9FCD8CD9+dkBt8VAJekCWVhEltcRHxlznbnk2jxm93xSD2o6gZ5X0VSaSUxyEhm
J+8F3gyTHGgbq/TgyjFoockWh5EtGgAFuWvmPJCF5P0/UaNeoKwGwSJBu6oTXkHx
R4nvVMRcj5UgTsKpZ79NiDQukbjG5ScNT5TCUiiZsBXBqBx3fD61EH6cAuh4P3Kr
iM7PY4fWAHo890Dx+Qlt
=WIhx
```

```
-----END PGP MESSAGE-----
```

```
--=-=01-e8k41e11ob31eefa36wo=-=-
```

The server decrypts this message to





Content-Type: application/pgp-keys

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
mDMEV2o9XRYJKwYBBAHaRw8BAQdAZ8zkuQDL9x7rcvvoo6s3iEF1j88Dknd9nZhL
nTEoBRm0G3BhdHJpY2UubHVtdW1iYUBleGFtcGxlLm5ldIh5BBMWCAAhBQJXaj1d
AhsDBQsJCAcCBhUICQoLAgQWAgMBAh4BAheAAAJEB0VY2gqAg0KmQ0BAMUNzALT
OzG7tolSI92lhePi5VqutdqTEQTyYYWi1aEsAP0YfiuosNggTc0oRTSz46S3i0Qj
AlpXwfU00888yIreDbg4BFdqPY0SCisGAQQB1UBBQEBB0AWeeZlz3104qTmIKr3
CZh1RUXZFxc3YKyoCXyIZBBRawMBCAeIYQQYFggACQUCV2o9jQIbDAAKCRATlwNo
KgINCsuFAP9BplWl813pi779V80MsRGs/ynyi hn0ESft/H8qlM8PDQEAqIUPpIty
OX/OBFy2RI1Ii7J1bTp9RzcbzQ/4Fk4hwQQ=
=qRfF
```

-----END PGP PUBLIC KEY BLOCK-----

and returns this confirmation request



From: key-submission@example.net  
To: patrice.lumumba@example.net  
Subject: Confirm your key publication  
MIME-Version: 1.0  
Content-Type: multipart/encrypted;  
          protocol="application/pgp-encrypted";  
          boundary="--=01-wrzqued738dfx4x97u7y=-="

Date: Wed, 05 Oct 2016 10:16:57 +0000

--=01-wrzqued738dfx4x97u7y=-=  
Content-Type: application/pgp-encrypted

Version: 1

--=01-wrzqued738dfx4x97u7y=-=  
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----

hF4DkYWHjk/NdMASAQdAluQeqhECpU2T0zEyBAEbFzhLkpubN160wjFCrtUc0Mw  
FwYgm2fp9cvTMDJ/xjkvmAcIEOT4AY/hn1yFQ4z0KG0gCkSac+8mkDylnPdxlXYw  
0sBSAXlbqpVA7eUpFuU2Zs10zbIXxlwe6osR5wUIJut/RC0sYQmfvx55x8mUX5/  
zgTnNz1Mzye5ws4pTgAeQm2x0Yv018L8IZgY5KxwJLBzlss0wLZ45ZcS80hR11Fx  
NCow1fKF8lMn0JxagTE0ih807nctz8vT5bR1gx0d7N3LM+th8nAg9/6Ghf1XTpLo  
MzwGW0FtOG7Dg1Uxbw2bja0uRBeh6IIpmNAw1pmIfnNu7PpoRydU5w1K/R8MT06z  
MKdJ7IW5mVGes9EGnG3e4mjuILvNaZhFYy+a73IhDSaPm3oqdl1Qx7tbNg6lGjn6  
KStCYAcPGPp3m7aWkfsPGTh0VRhEXqaFFyfwfSVEj1pdIRjDFA==  
=Cdjh

-----END PGP MESSAGE-----

--=01-wrzqued738dfx4x97u7y=-=-

The client decrypts the attachment as

Content-Type: application/vnd.gnupg.wks  
Content-Transfer-Encoding: 8bit

type: confirmation-request  
sender: key-submission@example.net  
address: patrice.lumumba@example.net  
fingerprint: B21DEAB4F875FB3DA42F1D1D139563682A020D0A  
nonce: f5psc57zj6fk11wekk8gx4cmrb659a7

creates this response



Content-Type: application/vnd.gnupg.wks  
Content-Transfer-Encoding: 8bit

type: confirmation-response  
sender: key-submission@example.net  
address: patrice.lumumba@example.net  
nonce: f5pscz57zj6fk11wekk8gx4cmrb659a7

and sends it encrypted to the server

From: patrice.lumumba@example.net  
To: key-submission@example.net  
Subject: Key publication confirmation  
MIME-Version: 1.0  
Content-Type: multipart/encrypted;  
          protocol="application/pgp-encrypted";  
          boundary="--=01-iacqg4og4pqz11a5cg1o=-="

Date: Wed, 05 Oct 2016 10:18:52 +0000

--=01-iacqg4og4pqz11a5cg1o=-=  
Content-Type: application/pgp-encrypted

Version: 1

--=01-iacqg4og4pqz11a5cg1o=-=  
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----

hF4DUgLY5tvmW2sSAQdAnB1C3PMjS4AsGU0qaCqBdWQ05i6blWEyZrEsY+JZY1Qw  
ooNq7zdVWOHhL9LPGAALAgOL3Qfz+dN2u5QamSQ/LJ2c8M0XipNs3lqlNH63yQN1  
0sAmAc3W8xkwul+rf60LK/gMi6WzM4fnUhd4D1LJGIJoNUN0l3636C7ec0t2lkMl  
5bVAYg/SyMT3ymyfQnvtiem2T5DSnPsS1g6n6QNXWvkqvX9yGxNsNDJEHTuGJB8k  
0JoRlFWQTEo6pgA89febWl1EdeM1pPLstQ2uZE8NPjXoY1nMxAlu+iPYsR41/4sg  
dqw0v5BPLh/GIat8hh9SPWCA9iKlgSQ/Eiv5DpjQogEzpriT55dkgfvSVYIACod0  
ShZ91YKkcZffevdY72omqTk10a1SUXehPooIlRFmroDsi3VDaRKrUIo=  
=7uve

-----END PGP MESSAGE-----

--=01-iacqg4og4pqz11a5cg1o=-=-

## [Appendix B](#). Changes Since -05

- o Add a submission address to the Policy Flags.
- o Add a suggestion to avoid creating an index file.



Author's Address

Werner Koch  
GnuPG e.V.

Email: [wk@gnupg.org](mailto:wk@gnupg.org)

URI: <https://gnupg.org/verein>