



# Autocrypt Level 1 Specification

*Release 1.0.0*

**Autocrypt team, licensed CC0**

**Jun 12, 2018**

Autocrypt aims to incrementally and carefully replace cleartext e-mail with end-to-end encrypted e-mail. This differs from the traditional approach of maximizing the security of individual mail communications. **Sometimes Autocrypt recommends to send cleartext mail even though encryption appears technically possible.** This is because we want to avoid unreadable mail for users. Users may mix both Autocrypt-capable and traditional mail apps and they may lose devices or in other ways the ability to decrypt in unrecoverable ways. Reverting to cleartext when we suspect such situations is a key part of our aim to stay out of the way of users.

Another major difference in approach is that Autocrypt Level 1 only defends against passive data collection attacks. We share and support **the new perspective stated in RFC7435 (“Opportunistic Security: Some Protection Most of the Time”)**<sup>1</sup>. Protection against active adversaries (those which modify messages in transit) is the aim of future specifications.

**Level 1 makes it easy for users to encrypt, based on an automatic and decentralized key distribution mechanism. There are no dependencies on key servers and it is meant to work with existing e-mail providers.** Level 1 focuses on the use of Autocrypt on a single device. Users get rudimentary support on using Autocrypt on more than one device or mail app. This is internally realized through sending and receiving an Autocrypt Setup Message, secured by manually entering a long number. Improving usability for maintaining synchronized Autocrypt state on multiple devices is the aim of future specification efforts.

**Last but not least, Level 1 is meant to be relatively easy for developers to adopt.** It describes the basic capabilities required for a mail app to be Autocrypt-capable at Level 1, allowing it to exchange end-to-end encrypted e-mails with other Autocrypt-capable mail apps. The spec contains detailed guidance on protocol, internal state and user interface concerns. We have a good track record of supporting new implementers. Please don't hesitate to **contact the group**<sup>2</sup> or bring up issues or pull requests. Autocrypt is a living specification and we envision both bugfix and backward-compatible feature releases.

---

<sup>1</sup> <https://tools.ietf.org/html/rfc7435.html#section-1.2>

<sup>2</sup> <https://autocrypt.org/en/latest/contact.html>

# Contents

<b>1 Terminology</b>	<b>3</b>
1.1 Keywords to indicate requirement levels . . . . .	3
<b>2 Overview</b>	<b>3</b>
2.1 Approach and High Level Overview . . . . .	3
2.2 Requirements on MUA/E-mail Provider interactions . . . . .	3
2.3 Autocrypt Internal State . . . . .	4
<b>3 Peer State Management</b>	<b>5</b>
3.1 The Autocrypt Header . . . . .	5
3.2 Internal state storage . . . . .	6
3.3 Updating Autocrypt Peer State . . . . .	7
3.4 Provide a recommendation for message encryption . . . . .	7
3.5 Mail Encryption . . . . .	9
3.6 Key Gossip . . . . .	10
<b>4 Managing accounts controlled by the MUA</b>	<b>11</b>
4.1 Secret key generation and storage . . . . .	11
4.2 Handling Multiple Accounts and Aliases . . . . .	11
4.3 Avoiding MUA Conflicts . . . . .	12
4.4 Autocrypt Setup Message . . . . .	12
<b>5 User Interface</b>	<b>15</b>
5.1 Message Composition . . . . .	15
5.2 Account Preferences . . . . .	15
5.3 Helping Users get Started . . . . .	15
5.4 Disabling Autocrypt . . . . .	16
5.5 Destroying Secret Key Material . . . . .	16
<b>6 Appendix</b>	<b>16</b>
6.1 E-mail Address Canonicalization . . . . .	16
6.2 Example Autocrypt headers . . . . .	17
6.3 Example Autocrypt Gossip headers . . . . .	17
6.4 Example Copy when a Reply can't be Encrypted . . . . .	22
6.5 Example User Interaction for Setup Message Creation . . . . .	22
6.6 Example User Interaction for Setup Message Receipt . . . . .	22
6.7 Example Setup Message . . . . .	22

---

# 1 Terminology

## 1.1 Keywords to indicate requirement levels

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in the IETF’s Best Current Practice 14 (as defined in [RFC 2119](https://tools.ietf.org/html/rfc2119)<sup>3</sup> and [RFC 8174](https://tools.ietf.org/html/rfc8174)<sup>4</sup>) when, and only when, they appear in all capitals, as shown here.

## 2 Overview

### 2.1 Approach and High Level Overview

Autocrypt’s primary goal is to automate both secret and public key management so that users can encrypt mail without specialized knowledge.

This specification adds an *Autocrypt-specific mail header* (page 5) to outgoing mails, which contains, among other information, the sender’s public key. Transferring public keys in-band means that key discovery in Autocrypt does not require external infrastructure like OpenPGP key servers or x509 PKI.

Autocrypt provides a *set of rules* (page 7) that tracks this information for each communication peer. Autocrypt uses this information to determine whether encryption is possible and makes a *recommendation* (page 7) about whether encryption should be enabled for a given set of recipients.

This specification also introduces the *Autocrypt Setup Message* (page 12) as a way to transfer secret key material and related settings to other e-mail programs controlled by the same user. This spec also provides guidance on how and when to *generate* (page 13), *look for* (page 15), and *import* (page 14) these messages.

Autocrypt aggressively distributes public keys, but conservatively recommends encryption to avoid disruption to established email workflows. Specifically, Autocrypt only recommends that an email be encrypted if encryption is possible, and:

1. The sender specifically requests encryption during message composition;
2. The email is in reply to an encrypted message; or,
3. The sender and the recipients have explicitly stated that they *prefer* (page 5) encrypted email.

### 2.2 Requirements on MUA/E-mail Provider interactions

Autocrypt tries to impose minimal requirements on MUA and e-mail service interactions. Specifically, an Autocrypt-capable MUA needs to be able to:

- Control the contents of outgoing e-mail including the ability to set custom e-mail headers;
- Send e-mail on its own (required by the *Autocrypt Setup Message* (page 12));
- Read whole, raw e-mails including message headers; and,
- Optionally, scan the user’s mailbox for mail with specific headers.

If a particular e-mail account does not expose one of the required features (e.g., if it only exposes a javascript-driven web interface for message composition that does not allow setting e-mail headers), then the e-mail account cannot be used with Autocrypt. An Autocrypt-capable MUA may still access and control the account, but it will not be able to enable Autocrypt on it.

---

<sup>3</sup> <https://tools.ietf.org/html/rfc2119>

<sup>4</sup> <https://tools.ietf.org/html/rfc8174>

## 2.3 Autocrypt Internal State

An Autocrypt MUA needs to associate information with the peers it communicates with and the accounts it controls.

### 2.3.1 Communication Peers

Each communication peer is identified by an e-mail address. Autocrypt associates state with each peer. Conceptually, we represent this state as a table named `peers`, which is indexed by the peer’s *canonicalized e-mail address* (page 16), .

For the peer with the address `addr`, an MUA MUST associate the following attributes with `peers[addr]`:

- `last_seen`: The UTC timestamp of the most recent effective date (*definition* (page 7)) of all messages that the MUA has processed from this peer.
- `autocrypt_timestamp`: The UTC timestamp of the most recent effective date (the “youngest”) of all messages containing a valid Autocrypt header that the MUA has processed from this peer.
- `public_key`: The value of the `keydata` attribute derived from the youngest Autocrypt header that has ever been seen from the peer.
- `prefer_encrypt`: The `prefer-encrypt` value (either `nopreference` or `mutual`) derived from the youngest Autocrypt header ever seen from the peer.

Autocrypt-capable MUAs that implement *Gossip* (page 10) should also associate the following additional attributes with `peers[addr]`:

- `gossip_timestamp`: the UTC timestamp of the most recent effective date of all messages containing a valid Autocrypt-Gossip header about the peer.
- `gossip_key`: the value of the `keydata` attribute derived from the most recent message containing a valid Autocrypt-Gossip header about the peer.

How this information is managed and used is discussed in *Peer State Management* (page 5).

### 2.3.2 Accounts controlled by the MUA

A Level 1 MUA maintains an internal structure `accounts` indexed by the account’s *canonicalized e-mail address* (page 16) (`addr`). For each account controlled by the MUA, `accounts[addr]` has the following attributes:

- `enabled`: a boolean value, indicating whether Autocrypt is enabled for this account.
- `secret_key`: The RSA secret key material used for the account (see *Secret key generation and storage* (page 11)).
- `public_key`: The OpenPGP transferable public key (**OpenPGP “Transferable Public Key”**<sup>5</sup>) derived from the secret key.
- `prefer_encrypt`: The user’s encryption preference for this account. This is either `mutual` or `nopreference`. This SHOULD default to `nopreference`.

If `accounts[addr].enabled` is `true`, the MUA SHOULD allow the user to switch the setting for `accounts[addr].prefer_encrypt`. This choice might be hidden in something like a “preferences pane”. See *Account Preferences* (page 15) for a specific example of how this could look.

How this information is managed and used is discussed in *Managing accounts controlled by the MUA* (page 11).

<sup>5</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.1>

## 3 Peer State Management

An Autocrypt MUA updates the state it holds for each communication peer using the e-mails received from that peer. Specifically, Autocrypt updates the state using the `Autocrypt` e-mail header.

### 3.1 The Autocrypt Header

The `Autocrypt` header has the following format:

```
Autocrypt: addr=a@b.example.org; [prefer-encrypt=mutual;] keydata=BASE64
```

There are three defined attributes:

- The `addr` attribute is mandatory, and contains the single recipient address this header is valid for. If this address differs from the one in the `From` header, the entire `Autocrypt` header **MUST** be treated as invalid.  
**The Internet Message Format**<sup>6</sup> documents three types of originator fields: `From`, `Sender`, and `Reply-To`. Autocrypt is concerned only with the `From` field, and ignores the other originator fields.
- The `prefer-encrypt` attribute is optional and can only occur with the value `mutual`. Its presence in the `Autocrypt` header indicates an agreement to enable encryption by default with other peers who have the same preference. An Autocrypt Level 1 MUA that sees the attribute with any other value (or that does not see the attribute at all) should interpret the value as `nopreference`.
- The `keydata` attribute is mandatory, and contains the key data for the specified `addr` recipient address. The value of the `keydata` attribute is a Base64 representation of the binary **OpenPGP “Transferable Public Key”**<sup>7</sup>. For ease of parsing, the `keydata` attribute **MUST** be the last attribute in this header.

Additional attributes are possible before the `keydata` attribute. If an attribute name starts with an underscore (`_`), it is a “non-critical” attribute. An attribute name without a leading underscore is a “critical” attribute. The MUA **SHOULD** ignore any unsupported non-critical attributes and continue parsing the rest of the header as though the attribute does not exist. It **MUST** treat the entire `Autocrypt` header as invalid if it encounters a “critical” attribute that it doesn’t support.

To introduce incompatible changes, future versions of Autocrypt may send multiple Autocrypt headers, and hide the incompatible headers from Level 1 MUAs by using critical attributes. According to the above rules, such headers will be judged invalid, and discarded by level 1 MUAs. Such an update to the specification will also have to describe how an updated MUA will deal with multiple valid headers.

#### 3.1.1 OpenPGP Based key data

The `keydata` sent by an Autocrypt-enabled Level 1 MUA **MUST** consist of an **OpenPGP “Transferable Public Key”**<sup>8</sup> containing exactly these five OpenPGP packets:

- a signing-capable primary key
- a user id
- a self signature over the user id by the primary key
- an encryption-capable subkey
- a binding signature over the subkey by the primary key

The content of the user id packet is only decorative. By convention, it contains the same address used in the `addr` attribute placed in angle brackets. (This makes it conform to the **RFC 5322**<sup>9</sup> grammar `angle-addr`.) For compatibility concerns, the user id **SHOULD NOT** be an empty string.

These packets **MUST** be assembled in binary format (not ASCII-armored), and then base64-encoded.

<sup>6</sup> <https://tools.ietf.org/html/rfc5322.html#section-3.6.2>

<sup>7</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.1>

<sup>8</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.1>

<sup>9</sup> <https://tools.ietf.org/html/rfc5322.html>

A Level 1 MUA MUST be capable of processing and handling 2048-bit and 3072-bit RSA public keys. It MAY support other OpenPGP key formats found in an Autocrypt header (for example, by passing it agnostically to an OpenPGP backend for handling).

### 3.1.2 Header injection in outbound mail

During message composition, if the `From:` header of the outgoing e-mail (the `from-addr`) matches an address for which `accounts[from-addr].enabled` is `true` and the Autocrypt-capable MUA has secret key material (`accounts[from-addr].secret_key`), the MUA SHOULD include an Autocrypt header.

This header MUST contain the corresponding public key material (`accounts[from-addr].public_key`) as the `keydata` attribute, and `from-addr` as the `addr` attribute. The most minimal Level 1 compliant MUA will only include these two attributes. If `accounts[from-addr].prefer_encrypt` is set to `mutual`, then the header MUST have a `prefer-encrypt` attribute with the value `mutual`.

The MUA MUST NOT include more than one valid Level 1 Autocrypt header (see *Updating Autocrypt Peer State* (page 7)).

If the `From` address changes during message composition (e.g., if the user selects a different outbound identity), then the MUA MUST change the Autocrypt header accordingly.

An MUA SHOULD send out the same `Autocrypt:` header in all messages from a given outbound identity. An MUA SHOULD NOT vary the header based on the message's recipients. If (for whatever reason) the MUA needs to update (or discovers an update of) the user's `keydata` at some point, the MUA SHOULD send the updated `keydata` in all subsequent Autocrypt headers.

See *Example Autocrypt headers* (page 17) for examples of outbound headers and the following sections for header format definitions and parsing.

## 3.2 Internal state storage

See *Communication Peers* (page 4) for the information stored for each communication peer.

Autocrypt MUAs keep state about each peer, to handle several nuanced situations that have caused trouble or annoyance in the past. This state is updated even when the peer sends mail without an Autocrypt header.

For example, if a remote peer disables Autocrypt or drops back to only using a non-Autocrypt MUA, we must stop sending encrypted mails to this peer automatically.

In addition to the per-peer state described in *Communication Peers* (page 4), MUAs MAY also store other information gathered for heuristic purposes, or for other cryptographic schemes (see [the Autocrypt website](#)<sup>10</sup> for some example ideas).

However, in order to support future synchronization of Autocrypt state between MUAs, it is critical that Autocrypt-capable MUAs maintain the state specified here, regardless of what additional state they track.

---

#### Note:

- An implementation MAY also choose to use keys from other sources (e.g., a local keyring) at its own discretion.
- If an implementation chooses to automatically ingest a key from an `application/pgp-keys` attachment as though it was found in an Autocrypt header, it should only do so if the attached key has a **User ID**<sup>11</sup> that matches the message's `From` address.

---

<sup>10</sup> <https://autocrypt.org/en/latest/optional-state.html>

<sup>11</sup> <https://tools.ietf.org/html/rfc4880.html#section-5.11>

### 3.3 Updating Autocrypt Peer State

Incoming messages may be processed to update the `peers` entry for the sender identified by `from-addr` as extracted from the `From` header, by an MUA at receive or display time.

Messages SHOULD be ignored (i.e., `peers[from-addr]` SHOULD NOT be updated) in the following cases:

- The content-type is `multipart/report`. In this case, it can be assumed the message was auto-generated. This avoids triggering a `reset` state from received Message Disposition Notifications (RFC 3798<sup>12</sup>).
- There is more than one address in the `From` header.
- The MUA believes the message to be spam. If the user marks the message as not being spam the message MAY then be processed for `Autocrypt` headers.

When parsing an incoming message, an MUA SHOULD examine all `Autocrypt` headers, rather than just the first one. If there is more than one valid header, this SHOULD be treated as an error, and all `Autocrypt` headers discarded as invalid.

Updating `peers[from-addr]` depends on:

- the `effective date` of the message, which we define as the sending time of the message as indicated by its `Date` header, or the time of receipt if that date is in the future or unavailable.

---

**Note:** A message without a `Date` header, or with a `Date` that seems to be in the far future can cause problems for MUAs that encounter the message repeatedly (e.g. re-delivery, subsequent scans, etc). An MUA MAY decide to ignore such a message entirely for the purposes of `Autocrypt` processing. If an MUA is capable of associating information with a received message, it could instead save the `effective date` of such a message the first time it sees it to avoid accidental re-processing.

---

- the `keydata` and `prefer-encrypt` attributes of the single valid `Autocrypt` header (see above), if available.

The update process proceeds as follows:

1. If the message's effective date is older than the `peers[from-addr].autocrypt_timestamp` value, then no changes are required, and the update process terminates.
2. If the message's effective date is more recent than `peers[from-addr].last_seen` then set `peers[from-addr].last_seen` to the message's effective date.
3. If the `Autocrypt` header is unavailable, no further changes are required and the update process terminates.
4. Set `peers[from-addr].autocrypt_timestamp` to the message's effective date.
5. Set `peers[from-addr].public_key` to the corresponding `keydata` value of the `Autocrypt` header.
6. Set `peers[from-addr].prefer_encrypt` to the corresponding `prefer-encrypt` value of the `Autocrypt` header.

### 3.4 Provide a recommendation for message encryption

On message composition, an `Autocrypt`-capable MUA can decide whether to try to encrypt the new e-mail message. `Autocrypt` provides a recommendation for the MUA.

All `Autocrypt`-capable MUAs should be able to calculate the same `Autocrypt` recommendation.

This recommendation algorithm provides sensible guidance that avoids many common problems, and `Autocrypt`-capable MUAs SHOULD follow the recommendation. An implementation that deviates from the recommendation should do so on the basis of specific external evidence or knowledge, while carefully considering the impact of any variation, including:

---

<sup>12</sup> <https://tools.ietf.org/html/rfc3798.html>

- does it increase the chance of producing unexpectedly unreadable mail (for either the sender or the recipient)?
- does it leak previously encrypted content in the clear?
- does it force the user to confront a choice they do not have the information or knowledge to make safely?

If an implementation deviates from the Autocrypt recommendation in a meaningful and useful way, the implementer should describe the variation publicly so it can be considered for future revisions of this specification.

### 3.4.1 Recommendation structure

The Autocrypt recommendation depends on the recipient addresses of the draft message, and on whether or not the message is a reply to an encrypted message. When the user changes the recipients during composition, the Autocrypt recommendation may change.

The output of the Autocrypt recommendation algorithm has two elements:

- `ui-recommendation`: a single state recommending the state of the encryption user interface, described below.
- `target-keys`: a map of recipient addresses to public keys.

`ui-recommendation` can take four possible values:

- `disable`: Disable or hide any UI that would allow the user to choose to encrypt the message. This happens iff encryption is not immediately possible.
- `discourage`: Enable UI that would allow the user to choose to encrypt the message, but do not default to encryption. If the user manually enables encryption, the MUA SHOULD warn that the recipient may not be able to read the message. This warning message MAY be supplemented using [optional counters and user-agent state](#)<sup>13</sup>.
- `available`: Enable UI that would allow the user to choose to encrypt the message, but do not default to encryption.
- `encrypt`: Enable UI that would allow the user to choose to send the message in cleartext, and default to encryption.

### 3.4.2 Recommendations for single-recipient messages

The Autocrypt recommendation for a message composed to a single recipient with the e-mail address `to-addr` depends primarily on the value stored in `peers[to-addr]` (page 4).

#### Determine if encryption is possible

If there is no `peers[to-addr]`, then set `ui-recommendation` to `disable`, and terminate.

For the purposes of the rest of this recommendation, if either `public_key` or `gossip_key` is revoked, expired, or otherwise known to be unusable for encryption, then treat that key as though it were `null` (not present).

If both `public_key` and `gossip_key` are `null`, then set `ui-recommendation` to `disable` and terminate.

Otherwise, we derive the recommendation using a two-phase algorithm. The first phase computes the `preliminary-recommendation`.

<sup>13</sup> <https://autocrypt.org/en/latest/optional-state.html>



## Preliminary Recommendation

If `public_key` is null, then set `target-keys[to-addr]` to `gossip_key` and set `preliminary-recommendation` to discourage and skip to the *Deciding to Encrypt by Default* (page 9).

Otherwise, set `target-keys[to-addr]` to `public_key`.

If `autocrypt_timestamp` is more than 35 days older than `last_seen`, set `preliminary-recommendation` to discourage.

Otherwise, set `preliminary-recommendation` to available.

## Deciding to Encrypt by Default

The final phase turns on encryption by setting `ui-recommendation` to `encrypt` in two scenarios:

- If `preliminary-recommendation` is either `available` or `discourage`, and the message is composed as a reply to an encrypted message, or
- If the `preliminary-recommendation` is `available` and both `peers[to-addr].prefer_encrypt` and `accounts[from-addr].prefer_encrypt` are mutual.

Otherwise, the `ui-recommendation` is set to `preliminary-recommendation`.

### 3.4.3 Recommendations for messages to multiple addresses

For level 1 MUAs, the Autocrypt recommendation for a message composed to multiple recipients, we derive the message's recommendation from the recommendations for each recipient individually.

The aggregate `target-keys` for the message is the merge of all recipient `target-keys`.

The aggregate `ui-recommendation` for the message is derived in the following way (the earliest matching rule encountered below takes precedence over later rules):

1. If any recipient has a `ui-recommendation` of `disable`, then the message's `ui-recommendation` is `disable`.
2. If every recipient has a `ui-recommendation` of `encrypt`, then the message `ui-recommendation` is `encrypt`.
3. If any recipient has a `ui-recommendation` of `discourage`, then the message `ui-recommendation` is `discourage`.

Otherwise, the message `ui-recommendation` is `available`.

While composing a message, a situation might occur where the `ui-recommendation` is `available`, the user has explicitly enabled encryption, and then modifies the list of recipients in a way that changes the `ui-recommendation` to `disable`. When this happens, the MUA should not disable encryption without communicating this to the user. A graceful way to handle this situation is to save the enabled state, and only prompt the user about the issue when they send the mail.

## 3.5 Mail Encryption

---

**Note:** An e-mail that is said to be “encrypted” here will be both signed and encrypted in the cryptographic sense.

---

An outgoing e-mail will be sent encrypted in either of two cases:

- the Autocrypt recommendation for the list of recipients is `encrypt`, and not explicitly overridden by the user, or

- the Autocrypt recommendation is available or discourage, and the user chose to encrypt.

When encrypting, the MUA **MUST** construct the encrypted message as a **PGP/MIME**<sup>14</sup> message that is signed by the user's Autocrypt key, and encrypted to the currently known Autocrypt key of each recipient, as well as the sender's Autocrypt key.

### 3.5.1 E-mail Drafts

For messages that are going to be encrypted when sent, the MUA **MUST** take care to not leak the cleartext of drafts or other partially composed messages to their e-mail provider (e.g., in the "Drafts" folder). If there is a chance that a message could be encrypted, the MUA **SHOULD** encrypt the draft only to itself before storing it remotely. The MUA **SHOULD NOT** sign drafts.

### 3.5.2 Cleartext replies to encrypted mail

In the common case, a reply to an encrypted message will also be encrypted. Due to Autocrypt's opportunistic approach to key discovery, however, it is possible that keys for some of the recipients may not be available, and, as such, a reply can only be sent in the clear.

To avoid leaking cleartext from the original encrypted message in this case, the MUA **MAY** prepare the cleartext reply without including any of the typically quoted and attributed text from the previous message. Additionally, the MUA **MAY** include some text in the message body describing why the usual quoted text is missing. An example of such copy can be found in *Example Copy when a Reply can't be Encrypted* (page 22).

The above recommendations are only "MAY" and not "SHOULD" or "MUST" because we want to accommodate a user-friendly Level 1 MUA that stays silent and does not impede the user's ability to reply. Opportunistic encryption means we can't guarantee encryption in every case.

## 3.6 Key Gossip

It is a common use case to send an encrypted mail to a group of recipients. To ensure that these recipients can encrypt messages when replying to that same group, the keys of all recipients can be included in the encrypted payload. This does not include BCC recipients, which by definition must not be revealed to other recipients.

The `Autocrypt-Gossip` header has the same format as the `Autocrypt` header (see *autocryptheaderformat* (page 6)). Its `addr` attribute indicates the recipient address this header is valid for as usual, but may relate to any recipient in the `To` or `Cc` header. See example in *Example Autocrypt Gossip headers* (page 17)

### 3.6.1 Key Gossip Injection in Outbound Mail

An Autocrypt MUA **MAY** include `Autocrypt-Gossip` headers in messages with more than one recipient. These headers **MUST** be placed in the root MIME part of the encrypted message payload. The encrypted payload in this case contains one `Autocrypt-Gossip` header for each recipient, each of which:

- **MUST** include an `addr` attribute that matches one of the recipients in the `To` or `Cc` headers.
- **MUST** include the `keydata` attribute which **MUST** contain the same public key which is used to encrypt the mail to the recipient referenced by `addr`. See also *Preliminary Recommendation* (page 9) for how this key is selected.
- If a key has multiple user ids, only one **SHOULD** be contained in `keydata`. This user id **SHOULD** be picked to match the `addr` attribute, if possible. This is only relevant for keys which came from or were merged with data from external sources.
- **SHOULD NOT** include a `prefer-encrypt` attribute.

To avoid leaking metadata about a third party in the clear, an `Autocrypt-Gossip` header **SHOULD NOT** be added outside an encrypted MIME part.

---

<sup>14</sup> <https://tools.ietf.org/html/rfc3156.html>

### 3.6.2 Updating Autocrypt Peer State from Key Gossip

An incoming message may contain one or more `Autocrypt-Gossip` headers in the encrypted payload. Each of these headers may update the Autocrypt peer state of the gossiped recipient identified by its `addr` value (referred to here as `gossip-addr`) in the following way:

1. If `gossip-addr` does not match any recipient in the mail's `To` or `Cc` header, the update process terminates (i.e., header is ignored).
2. If `peers[gossip-addr].gossip_timestamp` is more recent than the message's effective date, then the update process terminates.
3. Set `peers[gossip-addr].gossip_timestamp` to the message's effective date.
4. Set `peers[gossip-addr].gossip_key` to the value of the `keydata` attribute.

## 4 Managing accounts controlled by the MUA

See *Accounts controlled by the MUA* (page 4) for a definition of the structure of information stored about the MUA's own e-mail accounts.

### 4.1 Secret key generation and storage

The MUA SHOULD generate and store two RSA 3072-bit secret keys for the user, one for signing and self-certification, and the other for decrypting. An MUA with hardware constraints (e.g., one using an external crypto token) MAY choose to generate and store 2048-bit RSA secret keys instead. The MUA MUST be capable of assembling these keys into an OpenPGP certificate (RFC 4880 “Transferable Public Key”<sup>15</sup>) that indicates these capabilities.

#### 4.1.1 Secret key protection at rest

The secret key material should be protected from access by other applications or co-tenants of the device at least as well as the passwords the MUA retains for the user's IMAP or SMTP accounts.

The MUA MAY protect the secret key (and other sensitive data it has access to) with a password, but it SHOULD NOT require the user to enter the password each time they send or receive a mail. Since Autocrypt-enabled MUAs *sign all encrypted outgoing messages* (page 9), it could happen that the user has to enter the password very often, both for reading and sending mail. This introduces too much friction to become part of a routine daily workflow.

Note that password protection of the secret key carries with it a risk that the user might forget their password, which might result in catastrophic data loss. Unlike IMAP or SMTP credentials (which can be reset by the server operator given some sort of out-of-band confirmation), there is no recovery workflow possible for the loss of a password protecting a secret key. An MUA that chooses to offer password protection of the secret key (or other sensitive data) SHOULD support usable and secure backup/recovery workflows for the protected material.

Protection of the user's keys (and other sensitive data) at rest is achieved more easily and securely with filesystem-based encryption and other forms of access control.

### 4.2 Handling Multiple Accounts and Aliases

An MUA that is capable of connecting to multiple e-mail accounts SHOULD have a separate and distinct Autocrypt `accounts[from-addr]` for each e-mail account with the address `from-addr`.

A multi-account MUA MAY maintain a single `peers` table that merges information from e-mail received across all accounts for the sake of implementation simplicity. While this results in some linkability between accounts (the effect of mails sent to one account can be observed by activity on the other account), it provides a more uniform

<sup>15</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.1>

and predictable user experience. Any linkability concerns introduced by Autocrypt can be mitigated by using a different MUA for each e-mail account.

Sometimes a user may be able to send and receive emails with multiple distinct e-mail addresses (“aliases”) via a single account. For the purposes of Autocrypt, the MUA SHOULD treat each specific alias as a distinct account.

### 4.3 Avoiding MUA Conflicts

If more than one Autocrypt-enabled MUA generates a key and then distributes it to communication peers, encrypted mail sent to the user is only readable by the MUA that sent the last message. This can lead to behavior that is unpredictable and confusing for the user.

See section *Helping Users get Started* (page 15) for guidance on how to detect and avoid such a situation.

### 4.4 Autocrypt Setup Message

To avoid “lock-in” of secret key material on a particular MUA, Autocrypt level 1 includes a way to “export” the user’s keys and her *prefer-encrypt state* (page 4) for other MUAs to pick up, asynchronously and with explicitly required user interaction.

The mechanism available is a specially-formatted e-mail message called the Autocrypt Setup Message. An already-configured Autocrypt MUA can generate an Autocrypt Setup Message, and send it to itself. A not-yet-configured Autocrypt MUA (a new MUA in a multi-device case, or recovering from device failure or loss) can import the Autocrypt Setup Message and recover the ability to read existing messages.

An Autocrypt Setup Message is protected with a *Setup Code* (page 13).

#### 4.4.1 Message Structure

The Autocrypt Setup Message itself is an e-mail message with a specific format. While the message structure is complex, it is designed to be easy to pack and unpack using common OpenPGP tools, both programmatically and manually.

- Both the To and From headers MUST be the address of the user account.
- The Autocrypt Setup Message MUST contain an `Autocrypt-Setup-Message: v1` header.
- The Autocrypt Setup Message MUST have a `multipart/mixed` structure, and it MUST have as first part a human-readable description about the purpose of the message (e.g. `text/plain` or `text/html` or `multipart/alternative`).
- The second mime part of the message MUST have `Content-Type application/autocrypt-setup`, and SHOULD have `Content-Disposition of attachment`. Its content consists of the user’s ASCII-armored secret key, encrypted within an ASCII-armored OpenPGP symmetrically-encrypted message. Specifically, this means a block delimited with `-----BEGIN PGP MESSAGE-----` and `-----END PGP MESSAGE-----`, which contains two OpenPGP packets: a **Symmetric-Key Encrypted Session Key**<sup>16</sup> followed by a **Symmetrically Encrypted Integrity Protected Data Packet**<sup>17</sup>.
- There MAY be text above or below the ASCII-armored encrypted data in the second MIME part, which MUST be ignored while processing. This allows implementations to optionally add another human-readable explanation.
- The encrypted payload MUST begin with an ASCII-armored **RFC 4880 Transferable Secret Key**<sup>18</sup>. All trailing data after the first ASCII-armor ending delimiter MUST be stripped before processing the secret key. The ASCII-armored secret key SHOULD have an `Autocrypt-Prefer-Encrypt` header that contains the current `accounts[addr].prefer_encrypt` setting.

<sup>16</sup> <https://tools.ietf.org/html/rfc4880.html#section-5.3>

<sup>17</sup> <https://tools.ietf.org/html/rfc4880.html#section-5.13>

<sup>18</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.2>

- The symmetric encryption algorithm used MUST be AES-128. The passphrase MUST be the Setup Code (see below), used with **OpenPGP's salted+iterated S2K algorithm**<sup>19</sup>.

#### 4.4.2 Setup Code

The Setup Code MUST be generated by the implementation itself using a **Cryptographically secure pseudorandom number generator (CSPRNG)**<sup>20</sup>, and presented directly to the user for safekeeping. It MUST NOT be included in the cleartext of the Autocrypt Setup Message, or otherwise transmitted over e-mail.

An Autocrypt Level 1 MUA MUST generate a Setup Code as UTF-8 string of 36 numeric characters, divided into nine blocks of four, separated by dashes. The dashes are part of the secret code and there are no spaces. This format holds about 119 bits of entropy. It is designed to be unambiguous, pronounceable, script-independent (chinese, cyrillic etc.), easily input on a mobile device and split into blocks that are easily kept in short term memory. For instance:

```
9503-1923-2307-
1980-7833-0983-
1998-7562-1111
```

An Autocrypt Setup Message that uses this structure for its Setup Code SHOULD include a `Passphrase-Format` header with value `numeric9x4` in the ASCII-armored data. This allows providing a specialized input form during decryption, with greatly improved usability.

As a further measure to improve usability, it is RECOMMENDED to reveal the first two digits of the first block in a `Passphrase-Begin` header, sacrificing about 7 bits of entropy. Those digits can be pre-filled during decryption, which reassures the user that they have the correct code before typing the full 36 digits. It also helps mitigate a possible type of phishing attack that asks the user to input their Setup Code.

The headers might look like this:

```
Passphrase-Format: numeric9x4
Passphrase-Begin: 95
```

If those digits are included in the headers, they may also be used in the descriptive text that is part of the Setup Message, to distinguish different messages.

#### 4.4.3 Setup Message Creation

An Autocrypt MUA MUST NOT create an Autocrypt Setup Message without explicit user interaction. When the user takes this action for a specific account, the MUA:

- Generates a Setup Code.
- Optionally, displays the Setup Code to the user, prompts the user to write it down, and then hides it and asks the user to re-enter it before continuing. This minor annoyance is a recommended defense against worse annoyance: it ensures that the code was actually written down and the Autocrypt Setup Message is not rendered useless.
- Produces an ASCII-armored, minimized **OpenPGP Transferable Secret Key**<sup>21</sup> out of the key associated with that account.
- Symmetrically encrypts the OpenPGP transferable secret key using the Setup Code as the passphrase.
- Composes a new self-addressed e-mail message that contains the payload as a MIME part with the appropriate Content-Type and other headers.
- Sends the generated e-mail message to its own account.
- Suggests to the user to either back up the message or to import it from another Autocrypt-capable MUA.

<sup>19</sup> <https://tools.ietf.org/html/rfc4880.html#section-3.7.1.3>

<sup>20</sup> [https://en.wikipedia.org/wiki/Cryptographically\\_secure\\_pseudorandom\\_number\\_generator](https://en.wikipedia.org/wiki/Cryptographically_secure_pseudorandom_number_generator)

<sup>21</sup> <https://tools.ietf.org/html/rfc4880.html#section-11.2>

A Level 1 MUA MUST be able to create an Autocrypt Setup Message, to preserve users' ability to recover from disaster, and to choose to use a different Autocrypt-capable MUA in the future.

#### 4.4.4 Setup Message Import

An Autocrypt-capable MUA SHOULD support the ability to find and import an Autocrypt Setup Message when the user has not yet configured Autocrypt (that is, when `accounts[addr].secret_key` is unset). An MUA in this state could look for such a message in several ways, including:

- If the user decides to enable Autocrypt for an account, and indicates to the MUA that an older MUA has already enabled Autocrypt on that account, the new MUA could ask the user to generate an Autocrypt Setup Message from the old MUA, and then wait (e.g., via **IMAP IDLE**<sup>22</sup>) for such a message to arrive.
- The MUA could proactively scan the account's mailbox for a message that matches these characteristics, and it could alert the user if it discovers one.

When looking for an Autocrypt Setup Message, the MUA may encounter messages that look similar to what it expects, but are not well-formed. If the MUA discovers an e-mail message that has the `Autocrypt-Setup-Message` header but its value is not `v1`, the MUA SHOULD ignore this message entirely.

When looking for an Autocrypt Setup Message, if the MUA discovers a message with the `Autocrypt-Setup-Message: v1` header with `To:` and `From:` headers matching an account controlled by the MUA, but the message's metadata and structure is not as expected, the MUA SHOULD alert the user that a malformed Setup Message has been found, and it SHOULD NOT offer to import the message.

If the MUA finds a good Autocrypt Setup Message, it should offer to import it to enable Autocrypt. If the user agrees to do so:

- The MUA prompts the user for their corresponding Setup Code. If there is a `Passphrase-Format` header in the outer OpenPGP armor and its value is `numeric9x4`, then the MUA MAY present a specialized input dialog assisting the user to enter a code in the format described above. If there is no `Passphrase-Format` header, or the value is unknown, then the MUA MUST provide a plain UTF-8 string text entry.
- The MUA should try decrypting the message with the supplied Setup Code. The Code serves both for decryption as well as authenticating the message. Extra care needs to be taken with some PGP implementations that the Setup Code is actually used for decryption. For example, [this is difficult to do correctly with GnuPG](#)<sup>23</sup>.
- If it decrypts, then the MUA SHOULD update `accounts[addr]` according to the contents of the decrypted message, as discussed in *Accounts controlled by the MUA* (page 4).

See *Example Setup Message* (page 22).

Since Level 1 only recommends looking for a Setup Message when `accounts[addr].secret_key` is unset, some Level 1 MUAs might not look for or handle Setup Messages for an already-configured account at all. If two such MUAs share an account, and both MUAs have somehow enabled Autocrypt on it independently without discovery of a Setup Message, they will have different secret keys. This situation is bad because it may lead to intermittently unreadable mail on either or both MUAs.

These simple implementations can both keep Autocrypt enabled and avoid new unreadable mail if the user manually synchronizes secret keys. To do this, the user must first *destroy their local secret key* (page 16) on one MUA. Afterwards, that MUA can begin looking for a Setup Message again. A more sophisticated implementation may offer a more user-friendly way to detect this situation and resolve it.

<sup>22</sup> <https://tools.ietf.org/html/rfc2177.html>

<sup>23</sup> <https://dev.gnupg.org/T3277>

## 5 User Interface

Ideally, Autocrypt users see very little UI. However, some UI is inevitable if we want users to be able to interoperate with existing, non-Autocrypt users.

### 5.1 Message Composition

If an MUA is willing to compose encrypted mail, it SHOULD include some UI mechanism at message composition time for the user to choose between encrypted message or cleartext. This may be as simple as a single checkbox.

If the Autocrypt recommendation is `disable` for a given message, the MUA MAY choose to avoid exposing this UI during message composition at all.

If the Autocrypt recommendation is either `available` or `encrypt`, the MUA SHOULD expose this UI with the *recommended default* (page 7) during message composition to allow the user to make a different decision.

If the Autocrypt recommendation is `discourage`, then the MUA SHOULD expose the UI in an unactive state. But if the user chooses to activate it (e.g., clicking on the checkbox), then the UI should display a warning to the user and ask them to confirm the choice to encrypt.

### 5.2 Account Preferences

Level 1 MUAs SHOULD allow the user to disable Autocrypt completely for each account they control (that is, to set `accounts[addr].enabled` to `false`). For level 1, we expect most MUAs to have Autocrypt disabled by default. See *Disabling Autocrypt* (page 16) for more details.

### 5.3 Helping Users get Started

This section provides recommendations for MUA implementations to help users start Autocrypt immediately after an account (with the address `addr`) was set up.

The MUA SHOULD scan the mailbox for messages sent by the user (wherever the messages might be) that show evidence of OpenPGP or Autocrypt usage. It is likely sufficient to only scan the messages sent during the last 30 days, as it is unlikely that the user used Autocrypt or OpenPGP actively if no such message was sent in the recent past.

From the set of all found sent messages, the MUA should determine the best action to take from the following list of choices. Earlier choices are better than later ones.

1. If an Autocrypt Setup Message was found:  
Start a setup process suggesting the user to import the setup message. If multiple Autocrypt Setup Messages are found, the most recent message should be preferred.
2. If a sent message with an Autocrypt header was found:  
Provide guidance for creating an Autocrypt Setup Message on the MUA that created the message.
3. If there is evidence of actively used OpenPGP software (for example if a secret key is available, some specific software is installed, etc.) or if encrypted mails are found:  
Inform the user about Autocrypt on <https://autocrypt.org/pgp-users>.
4. If no evidence for Autocrypt was found:  
Create a key with default settings and without a password in the background. Set your `accounts[addr].prefer_encrypt` to `nopreference` and start sending Autocrypt headers.



## 5.4 Disabling Autocrypt

Once Autocrypt is enabled for a given account (`accounts[addr].enabled` is set to `true`), the user might choose to disable it. By default, disabling should only set `accounts[addr].enabled` to `false`, and it **SHOULD NOT** destroy `accounts[addr].secret_key`. This preserves the user's ability to read old encrypted e-mails, as well as being able to read encrypted e-mails that arrive after the user has disabled Autocrypt.

The act of re-enabling Autocrypt after it was disabled **SHOULD** leave `accounts[addr].secret_key` and `accounts[addr].public_key` intact, so that the user continues using the same key.

## 5.5 Destroying Secret Key Material

When disabling Autocrypt for an account, a Level 1 MUA **MAY** offer the user an opportunity to also destroy the secret key material for that account. Since Autocrypt clients generally do not discuss secret keys with users, a MUA offering this choice should use a phrase like “destroy access to encrypted messages”, rather than referring to “keys” or “key material”.

A MUA that allows the user this opportunity **SHOULD** clearly indicate to the user that the destruction of this secret key material will leave them unable to read any new messages that arrive encrypted. A MUA that only retains the encrypted form of archived messages **SHOULD** also indicate to the user that previously-received encrypted messages will become unreadable as well. Note that for some users, this is a desirable feature: “destroy all messages” is an appropriate action to take in some circumstances.

If the user selects this option, the MUA **MUST** clear both `accounts[addr].secret_key` and `accounts[addr].public_key`.

# 6 Appendix

## 6.1 E-mail Address Canonicalization

To keep consistent state referring to different but practically equivalent writings of an e-mail address, a MUA **SHOULD** canonicalize e-mail addresses when comparing them (for example for using an e-mail address as an index key).

Canonicalizing the domain part (the part after the @): A MUA **SHOULD** canonicalize the domain part using **IDNA2008 Punycode conversion to ASCII**<sup>24</sup>.

Canonicalizing the local part (the part before the @): Autocrypt-capable MUAs that encounter a peer's e-mail address where the local part appears to be valid UTF-8 **SHOULD** canonicalize the local part by making it all lower-case using the “empty” locale (see **W3C's discussion on Case folding**<sup>25</sup> for more details).

**SMTP specifications**<sup>26</sup> say the local part is technically domain-specific, and byte-for-byte arbitrarily sensitive. In practice, nearly every e-mail domain treats the local part of the address as a case-insensitive string. That is, while it is permitted by the standards, `John@example.org` is very unlikely to deliver to a different mailbox than `john@example.org`.

An Autocrypt-capable MUA that is configured to use an account that has an e-mail address whose local part is not a valid UTF-8 string, or who cannot receive mail at the canonicalized form of their associated address **SHOULD NOT** enable Autocrypt on that e-mail account without an additional warning to the user.

Other canonicalization efforts are considered for later specification versions.

<sup>24</sup> <https://tools.ietf.org/html/rfc5891.html#section-4.4>

<sup>25</sup> [https://www.w3.org/International/wiki/Case\\_folding](https://www.w3.org/International/wiki/Case_folding)

<sup>26</sup> <https://tools.ietf.org/html/rfc5321.html#section-2.3.11>



## 6.2 Example Autocrypt headers

Alice sends Bob a simple, unencrypted e-mail message that lets Bob write back encrypted if Bob is using an Autocrypt-enabled MUA:

```
Delivered-To: <bob@autocrypt.example>
From: Alice <alice@autocrypt.example>
To: Bob <bob@autocrypt.example>
Subject: an Autocrypt header example using RSA 3072 key
Autocrypt: addr=alice@autocrypt.example; prefer-encrypt=mutual; keydata=
mQGNBFn+zzUBDADBo2D+WUbm3lN1lXtQTxLhxVADIIMLKldFUgu5w1KAMrW0x9x27cRNxzVrTfiv
2FiwThUHZmJBFai8HtsMvn/svrCPeGPvkjTDMCWZaEEc5/g51Uyszjf6fUsGXsC9tUcva6pGHate
8Iwpz5stKjRKI3U/mPdQpXmaurwzEdvlnWNWni9Ao2rwwV+BK3J/98gBRFT8W6gv+T/YGXVrqXMoMM
KLTfze2uy00ExJkhI64upJzD0HUBGjElYdeSWz7lYhQ2y5cmnWPfrnOxiOCVyKrgBulksda5SIjE
qCJCvYprX/Wvh5feRXYftWVQUMeo6moNoHTM9X+zQJPWWuWivOJpamIuUCziEycX8RtRo0yAOPwc
/vIppoxAMusQCvNl5YwVECngzXUi3EB72wXJ4411VfzPCSlgVNZV7Yqx1lW4PMRCfB2obl025rk3
GDlmqEVcG1Hh4FtEBkmwVjiv4duN0E33r2Yf80sFAkKnRCR1lYn8409DaJGou41hEV+LAsUAQEQA
AbQyYTF1YmQ2OGQtOGM3Ny00NWI4LWIwMzMtOGNhyZnmN2QyMDZkQGF1dG9jcnlwdC5vcmeJAc4E
EwEiADgWIQTmBGjORnd8P86f0HJx28VlF95lpwUCWf7PNQIbAwULCQqHAgYVCAkKcWIEFgIDAQIE
AQIXgAAKCRBx28VlF95lp3C/C/9tthB5Q6oyyJERPZmRY3V8n60wd0h35uLqQfcb51UYKZ3j+61n
ckz2iB9LrRxY9Q3lWozMqza+Jze4/g/VYHLLS7Zg0M3pLKzbSEyDvZVT523BVFscQWjKq679JGZ/
xPzJOPablxdXfSKPEfNvzKgK+x0a4Q8b03SemL5mmGPBrnuCza/nFhevUrQbbtuUzhBnMFBsPKvz
WUTKHEgIDLqz+8auPOQZSbF2D/1BEvtbobdGQi+YJLaj77/pURR1kp7su51IffTs0qgMMJh8jwQY
lMQMhozy43eqT1y9QE+DH9RBAypcRCmTcBE5Z8apnWpH/axfCDjboWwD62gN0dawc7WEQ+rdgu8W
Tocoo4A6iyCk6Xs59mOGE0gsCdZvzKruJOYqvERzeDibDc3hXDjOE82okBjQhsOVCK3a7uyAIZnc
z9Kovi0CkQ9d3EuG8297HSf1/PupsiFgHBsJzmZ549+ZHLXlZ5ss4aj9Hpe7bCk8oUUL+A61+nNY
VsVDSO25AY0Ewf7PNQEMANI3/DkEjghl0SgsbzqHaUAohh+GSMXUD7dQn28ZGxR/2Y5wu705MdkP
MKIrsyQowSeGn18rnM1PxnRGQRX+QnVZTdk73VeMID6nM1TfV5gmkjcb6NphGPEOTZyJlBjgQxE
z2LUBhFLseRS/6COF5q6Tj+TJFSPbDs5kVm8LqAra2vdvdpxV69WP2FfzwhIKTzxEwnDKc3rp7yE
I52qz8xMTCO+IkBic9rwdj7TqJxMOTZQdfpy/ltiGwg3lCGYaHuejJzDQ1U/X60CEq/WT7/UVqNw
ZkrsT4uG9BFGW+WOXuOpgA4v0YQ62XQAotVNXUY10XFRsb6DTr6vYjd0Lk/z7icAX5uzjlfJN3TV
qJxS0pDwtfYD52B936+mizGR+97uyqEBVNQKw1pvKdZDruir4300k63TMO/4cAhXfw7q91/RMGg
TJX2UC/BGMiePziboP+GHX87hRmAvFCRjQc0KFyxJGbnKID3Kn/RhUrePCAVWI34lSQ0Do5qLlRn
9QARQAQABiQG2BBGBCAAgFIEE5gRozkTXfD/On9BycdvFZX/eZacFAln+zzUCGwwACgkQcdvFZX/e
ZaeaIwv/WR2LYKlPXe/1sMKfh+iSYeJjvqx15i4OaLumont+btZmpyYDU8sOaMB12oBgQ3sNYaQp
fkTk/QNw3lbuiROPJeANQzC7Ckjj3SDBFoMxyqxmzhH0P1qvT90VOB061P1aHg7usuU4+MuvLKrg
vaLtzK4xuiHIzpkTCvtcyNmiS5Qi2guPV32UQ6HccSIEaZO5w+z6a/V0JZ191VwOnOatUp4DsDHo
4KfcUKpNUKouUGgkOhLP7DmsqdlNqoKCw4PxnSsg7H5imHKF1Xo/8nh0G5W15kpJendiI1ZGy/yES
jn9i1kKSqL4X+R4PkT9foAootoK3TrLbcyHuxFj5umcUuqqGfsvjhgc/ZIyvvoRf4X0Bnn1h9hpo
6ZvBoPDM5lJxtUL64Zx5HXLd6CQXGfZfzVeM+ODqQyITGQT+p7uMDiZF42DKiTyJjJHABgiV+Jl6
IM4woaGfCwAU+0Vg+JDuf7Ec8iKx5UNDI18PJTTzGVp65Gvz2Mq/CHT/peFNHNqW
Date: Tue, 07 Nov 2017 14:53:50 +0100
Message-ID: <rsa-3072@autocrypt.example>
MIME-Version: 1.0
Content-Type: text/plain
```

This is an example e-mail with Autocrypt header and RSA 3072 key as defined in Level 1.

## 6.3 Example Autocrypt Gossip headers

After having received messages with Autocrypt headers from both Bob and Carol, Alice sends an e-mail to the two of them, with Autocrypt Gossip headers.

```
Delivered-To: <bob@autocrypt.example>
From: Alice <alice@autocrypt.example>
To: Bob <bob@autocrypt.example>, Carol <carol@autocrypt.example>
Subject: an Autocrypt Gossip header example
Autocrypt: addr=alice@autocrypt.example; prefer-encrypt=mutual; keydata=
mQGNBFn+zzUBDADBo2D+WUbm3lN1lXtQTxLhxVADIIMLKldFUgu5w1KAMrW0x9x27cRNxzVrTfiv
```

(continues on next page)

```

2FwThUHZmJBFai8HtsMvn/svrCPeGPvkjTDMCWZaEEc5/g51Uyszjff6fUsGXsC9tUcva6pGHaTe
8Iwpz5stKjRKI3U/mPdQpXmaurwzEdvlnWNni9Ao2rwVW+BK3J/98gBRFT8W6gv+T/YGXVrqXMoMM
KLTfze2uy00ExJkhI64upJzD0HUbGjElYdeSWz7lYhQ2y5cmnWPfrnOxiOCVYKrgBulksda5SIjE
qCJCvYprX/Wvh5feRXYftWVQUMeo6moNoHTM9X+zQJPWWuWivOJpamIuUCziEycX8RtRo0yAOPwc
/vIppoxAMusQCvN15YwVEcngzXUi3EB72wXJ4411VfzPCSlgVNZV7Yqx1lW4PMRcFB2ob1025rk3
GDlmgEVcG1Hh4FtEBkmVjiv4duN0E33r2Yf80sFAkKnRCR1lYn8409DaJGou41hEV+LAsUAEQEA
AbQyYTF1YmQ2OGQtOGM3Ny00NWI4LWIwMzMtOGNhYzNmN2QyMDZkQGF1dG9jcnlwdC5vcmeJAc4E
EwEiADgWIQTmBGjORnd8P86f0HJx28V1f951pwUCwf7PNQIbAwULCQgHAgYVCAkKcWIEFgIDAQIe
AQIXgAAKCRBx28V1f951p3C/C/9tthB5Q6oyyJERPZmRY3V8n60wd0h35uLqQfcb51UYKZ3j+61n
ckz2iB9LrRxY9Q31WozMqza+Jze4/g/VYHLlS7Zg0M3pLKzbSEyDvZVT523BVFscQWjkq679JGZ/
xPzJOPabludXfsKPEfNvzKgK+x0a4Q8b03SemL5mmGPBrnuCza/nFhevUrQbbtuUzhBnMFBsPKVz
WUTKHEgIDLqz+8auPOQZSbF2D/1BEvtbobdgQi+YJLaJ77/pURR1kp7su51IfftTs0qgMMJh8jwQY
lMQMhozy43eqT1y9QE+DH9RBAypcRCmTcBE5Z8apnWpH/axfCDjboWwD62gN0dawc7WEQ+rdgu8W
Tocoo4A6iyCk6Xs59mOGE0gsCdZvzKruJOYqvERzeDibDc3hXDjOE82okBjQhsOVCK3a7uyAIZnc
z9Kovi0CkQ9d3EuG8297HSf1/PupsiFgHBSJzmZ549+ZHLX1Z5ss4aj9Hpe7bCk8oUUL+A61+nNY
VsVDS025AY0Ewf7PNQEMANI3/DkEjghl0SgsbzqHaUAohh+GSMXUD7dQn28ZGxR/2Y5wu705MdkP
MKIrsyQowSeGn18rnM1PxnRGOx+QnVZTdk73VeMID6nM1Ttfv5gmkjcb6NphGPeOTZyJibjgQxE
z2LUbhFLseRS/6COF5q6Tj+TJFSPbDs5kVm8LqAra2vdvdpxV69WP2FfzWHIKTzxEwnDKc3rp7yE
I52qz8xMTCO+IkBic9rwdj7TqJxMOTZQdfpY/ltiGwg3lCGYaHuejJzDQ1U/X6OCEq/WT7/UVqNw
ZkrsT4uG9BFGW+WOXuOpgA4v0YQ62XQAotVNXUY10XFrSb6DTr6vYjd0Lk/z7icAX5uzj1fJN3TV
qJxS0pDwtfYD52B936+mizGR+97uyqEBVnQKw1pvKdZDruir4300k63TMO/4cAhXfw7q91/RMGg
TJX2UC/BGMiePziboP+GHX87hRmAvFCRjJ0cKfYxJGbnKID3Kn/RhUrePCAVWI341SQ0Do5qLlRn
9QARAQABiQG2BBgBCAAgFiEE5gRozkTXfD/On9BycdvFZX/eZacFAln+zzUCGwwACgkQcdoFZX/e
ZaeaIwv/WR2LYk1PXe/1sMKfh+iSYeJjvqx15i40aLumont+btZmpyYDU8sOaMB12oBgQ3sNYaQp
fkTk/QNw3lbuiROpJeANQzC7Ckj3SDBFoMxyqxmzhH0P1qvT90VOB061P1aHg7usuU4+MuvLKrg
vaLtzK4xuiHIzpkTCvtcyNmiS5Qi2guPV32UQ6HccSIEaZO5w+z6a/V0JZ191VwOnOatUp4DsDHo
4KfcUKpNUKouGgkOhLP7DmsqdlNqoKcW4PxnSsg7H5imHKF1Xo/8nh0G5W15kpJendi1lZGy/yES
jN9i1kKSqL4X+R4PkT9foAootoK3TrLbcyHuxFj5umcUuqqGfsvjhGc/ZlyvvoRf4X0Bnn1h9hpo
6ZvBoPDM51JxtUL64Zx5HXLd6CQXGfZfZVeM+ODqQyITGQT+p7uMDiZf42DKiTyJjJHABgiV+Jl6
IM4woaGfCwAU+0Vg+JDuf7Ec8iKx5UNDI18PJTtZGVp65Gvz2Mq/CHT/peFNHNqW
Date: Tue, 07 Nov 2017 14:56:25 +0100
Message-ID: <gossip-example@autocrypt.example>
MIME-Version: 1.0
Content-Type: multipart/encrypted;
  protocol="application/pgp-encrypted";
  boundary="PLdq3hBodDceBdiavo4rbQeh0u8JfdUHL"

--PLdq3hBodDceBdiavo4rbQeh0u8JfdUHL
Content-Type: application/pgp-encrypted
Content-Description: PGP/MIME version identification

Version: 1

--PLdq3hBodDceBdiavo4rbQeh0u8JfdUHL
Content-Type: application/octet-stream; name="encrypted.asc"
Content-Description: OpenPGP encrypted message
Content-Disposition: inline; filename="encrypted.asc"

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

hQGMAypihPateFlyAQv+Mnd0eKclm2/+RU4Qp3zmbQ3+5mHE7p3ZLiwnN7Xk7NXC
rqTEHpAQUDEYiXhs4tvmuDH7t+OG1kOPdfG66Cz1cLcWGrLI4AVC6Y5rBze1Ejo6
z3oFto3dmA4F1NTT8I8K6DYefzmlkuamKcsVTTagkVfx084w1NL1BYJbKnYkLbyt
Nfa6xfunYkvUCD8+ymwBzuPMwhFJt2EicFTTIHk1RSu2K+wC1ULx0hSluU+kMLWY
GW4DsMv1+TI8jQJNC1lMetjVwDrBSinKhbzj2bshhLFAQMPBLtrNu7QU+HmjDXrr
QrPgsW64veZe7hxChaqvQ3BAY9EML8+5KfR69AVHvkW5q+m20PPpKrjKhe7w4xj9
avJjSv8dmnNKONP YdgVL0NjyB6cjWFPq9f7ZjvUw1QIj3wuZS9msSt/8vU91+kq4
HOWLu/cME10r6X9osQjo4XesjJVJTTF35/XraSts5EE/R7VTOmqP/Nw5Y/VO9E1g
kl2nXAnEXVyIY/lv0BlghQGMA1T/aSXWYfnUAQv8dJk2YiuZa2Ky8NBmoxXp2ZHe
HIuSqXp8Efv0FMjCka3tFRr9mla1/e2DkCe/37VW/5kzBhBvUjXLDZ+tkiijw8cy
Gi9v1WinZdaAKyua05BU91cGd3YX2JeBvMsQqAPytz1BQPMtdqyyZQsU0hHqbpKJ

```

nNcrdfS3wX5qJg2DjtRpurjHiNvfLuWmizZ1ZvovXKB/WcpbjERU3XP7P×3sFMIB  
8gM8YeZoFG1GUAs8XFW1JjpNDMYX7CyG/wQGQHmNm64P6sNELN+2R6omV0xvalHx  
hHXfoYOnjs3AQ16xJoo42s4q+6Fc2PcCst8OsMO/ZZrPtEILNTG0RXId4ZF19Jwc  
rF/7xP0Bu3WamtnxO56IaxGpqqv2vRieaexmrWIT6Vy79qq+86G2OqV6oOfZX9FU  
bG2YJ77s/S4GveLuHTE/F5LqP/TShBLGxdpKFHoCES0HmvbJ2iXgmB6Xw1rfbzQu  
oOe3R919KoymHRdUsQ33b55cv21rWQt/z5cv82FFhQGMA4BmeZ3vRabVAQwArIzC  
JJvNYTPxX7wz/5IsORyKaYz/IL03hBbdar6ZTcdJ9J9en2vriaxYn4Z7Y1+N+a1Z  
i9ddTHCjPvV09X6M1LLXHF6hpnEn3Tafz2/pSpJYE38E8r8h6036Fkljq+C4fj  
dyTdcOVIdkbzlcGyRYQ4ysDlrSNRkLbcj7EJ31HZ3peA+YgCI0EwDPPUKXEfp96S  
xUpSHlxPlZ6V126/NS1o/j+d5VAQivCg9+J2oeIT9jM1uGfgur3p2SmES0oFSQJM  
eAcY1qLZimlORc01HRiXkX4bHO0C99W8qRy+Pj5EIVgI500FjsljlG6WbKMMab9  
odP/nULfxYeaTA+q06eZYXy/rHFVMTPr7SCR2LGfamptRtA6rfVA5uUrjUOtTTNg  
cpgWL5me7asAoy/vz+wDbjDsCI1pJ8M6r0PctusamMA+/QW80BbnXnwhFm/SSNeJ  
MfS+xcaKUu00M7JEyFLSE1at3Kp784ToCxfLSE7D5cd2LDkN2P4g9+K3Gli0uwB  
8VSq9pccsynDG/vozmTjGvLLhIoUBPqU03+zVq54osVmhsIS68evdTeDt21YZiBjG  
Ro4cmuqmpTJZFOzdCmzXbJsLiUDbJG1R8L+sn9Q5/FM3R5PTW38gl4y+atS++7Wj  
F95QR EAaPckLrCVVpT6AR6wGr4QjNehnb9Ykt2AcRZ/TcNqKt+P0QaDNoJrkMgld  
zuyxpq0NBjku7ZViRBSntDq3bdFpR76wCXPjnBjYYDtrX50aBE2MNwMgRPRkheP  
E67SEKODkNEzhx3IZP+xy5iQsAXnQKupLcKQ37vn2bLmd7YEAQvm3yvSWLw1ETNg  
eP+R3SktF8aZmmdwJnafQZDhWAaeXfhhA7wspWNkPKvWtG9gChYJbA5uKqC+oK+4  
U31wwwyjleV3JghC8MK/0md467WYiK7UOmEGAV0A+T8Ud0d23Qk1mBROtsftnWTW  
iP46TazPlyPe3T8XSk7pzb2zvrUM01WTktqSSnOL4pvwlKybtZMssKvRuwixZiJo  
bXvLvU8i/X4dRRIm6kvGF+AQ4xBCBHswUUsFb+T3Ljkrbi34pUEGeP+rIjj/DWi+  
ubtNlhjcNMxfs6Ropo6IsgihKTr7IdbuiXk98sxmRWrhfhk2BPWEU360Z1znDfe  
4rwrzJWLRic3Vf9I29o7CI0dwvglsdUqQRvY4gumiMHOFcm5Vic+DEwq4HWEYuV7  
r5sSs7j3WsuWvCUHETvFve4HeluhhvX9fjpHL8NyQeUfn1z26e08KN+fmBHRUbv6  
3fLYPw21hxW/2usilmLa52COi7XQaOuFdZZsIOON74ocSXf6UzLiQDeE89S2573P  
PkA6LiPTi6KrqaZzHC4IjmInPvr6SmxZ78g1Kshz/KUFczP Xu9frOn6uhatUb9bS  
SmFiuFoH+DLNQt19Ex/7iyTce7PjPVAiikqssDV43TwqsS25ncQ+ZPQIwrJyQL9  
SRYUiDZs6JQLvr15u1qxJslUPMJFauGxoyJWn9PoOqC6Lbh6Rf0kyezDa8DBaOL5  
mL7WNbm7zOHleuAwYCWKGu4oZz02HivJyZjRwo8LqCsQgt2hfMZAjq5VFYjNfOWT  
9CFmsX2nLQnJHMUrU3QIRqe/4HJrGGi3t+75aGZdehG+bfJQNMAa21uiu3V7r61s  
gE0ocrq34SUiIYN54sdW7BZsQYtgwJNEGUDFTQIjYoqC7GoG041H4U8NfdrQGCL  
tRE23v0HDUtFPq78nnLP4VooVtby0jcamf7v6EPx7uC0mBAa5hkBOgEoW/zguEm  
EB7Qqo8PmfFvE3hgDRaDvag2Yre9A9xXYXffmfccokBUk41yoW8hFUzMjphOuiV/S  
mPmK7Ztal544VPBVuJAcelQZ5cmfL0oprMjZd1jTltXYZxnzsAwCgRY8LBZC3tUT  
1MhXLRUCsGvLQDZYaCCz3dVwCyNH61N2vDN6CCU6LHTjgKCs0j0RfWjggGLH4gV  
BfSftlDYhd3bE8oLgqKkJTAWq8zcbU0xWuuBUJdWV3aGZ/nfc11ifxmOe4HQ2tXR  
tsFThrgOrtoaqfoTkUZxOpdpCya03d9nS/sahBtbw9vgaTKZVhBquvYV62V2R4dj  
oLH1YrverwfvTheCqmmiNMfdfsfxj+7q01gFV9iDVGkkg/HfoqTq1tsBpcXYmf9VX  
WN0x72/I7vJc9EV1K7411M+3KR+5/ikOK8/9ZFS0T4CDd1hPfdPxB2Vp5wu+Br12  
AubRHM6c3LYgeqtTjKZuIW8SjPLwW9RDXGn3tFh7faHRB2v/z7VBWSZ834ZrFPoZ  
61v523AEzyxkcLoyfDK8K1GBakfyUa j0Dui3fnWYzf06VfcVpjkxftz0e010HV+n  
Tt/GwaQVrwiUv8DPKaIwMKmhQnqSEtX3+BoxzE7UVJKNpKO82Ebfb6npP+A9qXbE  
ie2tjssJ5UMt3AgOZGJf85u7xvdqX+tUxLtK8ArL3mc8arWBa4GzaBHFqfW  
bq13FMk8FOQxIRUo3eItmOSLrKmyx51+31bcyzJp7e5BeCeq7fighwvX8dDdrzLy  
oR9oTHIEVT3cbzsqCVx9U09zuoL/eY3Girja6EBQJvs6KdTLz6LUx1fbKymQyJ+D  
uKdZNN82d6VPSwlapJvgvAmefyMO4uDyGGO2IT1FUBzVqFNq5h79EMFSglNDiLx  
KIOEXxcLuD+TWAngKbbZvY50qUh7fyNH2r0ic+D8X9nC0chCcVXe8vPV7Fjp2Uwh  
Kjqx9c0WDMkegqK54pcaU50KgQpqiGrfeMv+ALepDMwFJOpty85pK+n1HJi2gEqx  
NbVwAPkcpj7jV0e1KOGN5rDa7038nmlsIiDrDT84otK08KKaxtrMPC7nRrdODSZH  
NK+0tjmt7e0PtqH8D0i6BIZvBq4gH1kcOoPmECFMF/NdpCYLt1OPCCTtotKxUWL/  
EZ8ZnwxQkMebJrbWf14b+fcu40t1pkzH8uHpcDtNG4j4/uy01+MzvPsYNNbJJbc  
0Qogy2T2Ls7YInwcuNrYdVUECRNV4LjefCOHKiI/f/54ViJCDtDLnwrIHg5GFKoz  
SE+MLpP9X+hfnA0BkaAf10XmfKNms2RoI90nnAw830pb4o2cJAWGjhfYJpr3PGlo  
zrk8fZ0gBz4o0z4bEu3SHntYPyNaLR1bJY8Jx3FUM2N+ j1pt2yKf6J055PJgvbA+  
33MXJFPLcLjlx3Bhw6MfPc/F0ELUaIKpDdVYUWaoosb1NKse7wJIMvxwEhidxoK  
B0+d1M8uY6HgS6vCP3URL4YUW6yuHXRIq3d708d9iITSXeHz4RgraUvReC3sNv6i  
ScrZQx1uNt7Em9WHMVIInRWwff77smEgVw027/5LLrdve943MUDvREYPJYeMCeMul  
UQpHCZa2+RPq9X7LEzP2k1ge75uzCjPDatqvl1Ro3NnXtZ8CxXHwacoH1hV3Zbi7  
PmI2/V1wrj8gZewuLXShaIDUHSXypFoGig6jIaU6WPYKGFHMTkmTEFPfopPz10yY

(continued from previous page)

```

kVH4KbWXcw2mCMElet3tgAiVe4mgeJmULwe0UjfiBFzJuzth9dVKj48I7YFGXpM7
uevgFrh/Z1/y70sbaJdza3ZDOLt+qEAMRAcTPmuS9i4lafG2sLw9MXvHELTa2NZ1
eN/vx0hQeJFgAZMGKJev5GpMoEGKRNr2sk5ghv27y9trIqH6FrayFF/qTtQBQ+9
xGTtaLVCVCFpYahfFDSZ2kd4gsMlKHTQB2XNwB4h9p4eQn3ijNb3kdisK8sxp932
ArLUe9C2kdvega+zub9M6wbibYir+1653ojiDDr5130ZdShDS0VncqeyPZb9XPr4
jGzc8zWvp/z8kgTZXLonwteBc8MJ4jSZcd6CEqLKmUOsFQqA3lZiFvyZmq6rfmm0
Yt/yoP8EaOrh19dU9JqPffFh6+UNUGaf2+OL5kmU53bQk1DaQ79XRWP0FINVzrs1
ulz5ZW3tInTo/rkYDYGpWOXkkuW9PPnDN9J5yxmepfVxac86LVmMy5iK9gavjWAB
owk2pB2zm9ETu0Ac/piKmNicgjIcxZPy1JPTx1siZpGPNVv3v57KVBsNlpIuevQj
Q2uY6OUe7118Y5cWlGm7sid6vpLfsZqqR48OY4ZPt5qtpAl8pHan4RnqbEkCr9Di
vgZ+vTe8/v7n10NSAJHAbz4YME2JDbVDDZZRSecq1kpIBlwYXplPbGnpPjsTJu8y
+9+KLlV3dAHM9Hxhffs4qx8seJhCVzd9WyDBG13HH06Y2rLcbhyD4DEcvlUOXieN
tzwGF4RxmCDDGLJDYOz+4FAi0wADVuzLPXtbfSlw3jVtAmIk8A5rhwOGU01ePiL
NcaLhiAM6JLDEV/lykVuMvXQEVSV5vT3MovOWl/v3R9ve6+beGYzptJ/15oSooy
IK08XK1YyYnGDMEEnNOYfobCmw+/ctNPwdM5ioWskzUx6ku34G74049gtRccHnYa
uU715VvdDfRwrsrhNyMpVK9IacamzigskP/SXGzxDCK/jvN/3mc2X9U00JqhewdG
TgajsJr7AFwvj0yXlGsY9SvnaBosQqnvddD0dpvVvQXsLwXHCsngI/3XwFt15c3u
otHFmzijsm0+JczT5YfOqF7ZYst9Kb62G3MaSF0ymPaSppliyiDHZH1rVEQdx/+Tx
u+VKA/1SGxLvxcrdIinwlAcDRXAw7XNDmiYaOgOP6dRXF+4U1ysCS78WvvDfmMvB
PFoSPoozuKX6YbIUldrYBu+zQXIgAEUfecGwpqATWS7vXw29bLmvtLDh/rT72KsK
rRXBnLZenyDrJn0ra7relgWCwm80zhOeahzZ+vOZCQOz1R5hOG/A0EVwNAYEm3Kz
VKkKBjGRa3pmA9n4eLQFI4BVAPYQ5Wu+dWDBNu12HVGMAilIGrBYKh3VGTywpdYX
EhN/6BrTdcKnq2FgviH77ILee1p2RP+8ggdS65kdIFpYSSe0ticjL/58wwJXk11
yOOP66h60cG9i5rWOMl7m4KqhnGA8Nzivm38ItDTHPaUKYXsbcXF2d0lCzY8MIg
VoIALICwXlJ91n6MEN2dxIsCYECzJlGfHoKOMRueT0XkIBU1zRfGnS9Qyq7yBY02
jl/v/H8b6tf5oSa/uFblqccq9K4WjOSAH/vQaDmZpHx3xaUg2v2r0mepRVN3HAJVM
vFaFanOJeadCkFzLBrFRd0l/n3eStm9Gjiz6YW+ubPkfe2VU5zLyixkbg+LHUQs
X1rtdn6Kk5mgEC09fgtpCR6bdWgpRWV3AVShTktvY++EdMK6cWw9lawag5LRX4C2
x5R4tB9nuSYOowwWaKYZDQNmWTTklfGp4xoIOUeTDAPQm+yJR+1wLazHfTavVE89
Z3eEWJQKJUgeKe/bWzpfmwm6x7Zkx1OgHHQa6I3lqemNNwXRjXUGxFTThkiFv5k1B
2hViz+a8bAC/2KZlHtF6gPgVv74SiG2DtCsJHv/MRFgUpPt4azoXGI15CvPqRG6Z
yr0ogGXlKlwNyGmuQUk3tDuypoBnurW0w5pVch3p8DXlToUvYdClhAgMXZVVPDeH
+GNA8gk1wLn3uVQzHqUfJp/ZANmbtu89/kInq3FVbFQfl2OZmfavhwqLhONT08eU
CmhK8XVlP1BaMBZcMPLDC7dSF4mYsCkz3apDhrVAcWxJ2K06wV4aWj8I8C7B2Fmw
huM0JZ6yfBUTN/GcJs/NVXRA7HW/PuRktRB35Afp1C9w/LlDKNAAOostpIIG3oSL
xeIfgswUXwyKkSmoKg4wFtYLBfohhVfLsJmZQ4KYcYVN/yW09BwWvvljt1Y7a5Tm
j8Rth1RAHmCPLS8AlG+Nhw2DqMWzD4PNGKZw3ymrX63f96ay9aHZtTkKkrA809y1
7aHWVroj0/03p7boIDjx3hReACEgximebS3EgCw4MDB54+WcGRN7nsXJ3q/YAjH
iOmmaOjMNWZekp42ltKTWooCS4NbeZlcyb9Z5R4HSH3+VlLdrZW76aafTXN8nU5u
lsYXjzF7wSJurITquBoDxbdbU/toHt4VQ4Q3RcMb5HKVNS0TzeHnXibCGuJwJXK6
4rMh6nwQtEKPYyGxEG000jckjXuwRecF24K50maTwJf2s4Qsbjxu0XOUzCnSyk
Zse2TvX61sfjdJxlXZ2ouHZWYHPSOHaNX/CNrik5lL6XR52C5fCNY7FB7Krdlew1
5p31D4iR0JUub7x2jAbgiI9ALZGvw+4qdxXY4ifb94gpXNrn08MOs/MQn+q3sp6Pw
uARQ69T+4TIndTG5G6I5IMmK47+p/PqZW+kCX+T7mmDw/KKz10uKkxogifGU+mJb
4BVLtBj/V2HqyfjoF9rbE8kZzP0wxMf8I/1M29211Qmz7iLHwf0x50gX+fZIUks/
i09gfWywP+y3WtpNXtg1F5m2FtRRfUqK01f05I1LCVIOfo6Yt8UPJZ4KFROBn2Z9f
vA2yxe6xEaP3XJrDDrinz30dSEzJ9IYUub9kMxZowVwo+QtiJutkWJCOiX7btjBNT
EKjmyg==
=69xN
-----END PGP MESSAGE-----
--PLdq3hBodDceBdiavo4rbQeh0u8JfdUHL--

```

Since Alice encrypts messages to herself, the above message can be decrypted by her private key as well (see the *Example Setup Message* (page 22) for access to her private key)

When decrypted, the encrypted part contains:

```

Autocrypt-Gossip: addr=bob@autocrypt.example; keydata=
mQGNBFoBt74BDAC8AMs jPY17kxodbfmHah38ZQipY0yfu097WUBs2jeiFYlQdunPANi5VMgbAX+H
Qb8LBKkoUOmJQrOnj1EXz5ILEHc/rs1bJjdmCE8cw9X+EN6PW1y9XNxl0hR10GKjs9cVW87uPmz/

```

(continues on next page)

TkIslmfzVB6wJMI1l8ax/Kb3IKr6wQXUT+JvJaJWoVDOOD+7FPondqOxITXMEzinJtZqfEY6SB8q  
+ bwrP9bMSyGaJl0fxbqdUxU4iVj8b1JpxuhFtvZik8i06avrOPfYmSnqANBOECSmuC3Uf2p1IHJR  
Fd8o12j0lzdQQH3EACg4dAdIuGHZxdLBQSQ8o8HvBDno9epau142HDHKUji42mf9Bc9DK4wW6Szb  
BGdefacEmowrwnlRuc3TyFwfnLuM6AB3k0HhOfUz/4tFKWNlDxN/w6xT30GSE2pp21lu9xN1S6X  
GXOstmDX40tgdMIfZVZqzDWfKzLVeKsE7Z/SNcKous49FdipChdy0FuEi4ua3NBFb8ImK60AEQEA  
AbQyZjJlMDk5ZDQtYmM5Yi00MGm4LWE2NmItMWE1YmM0MjFjYjRhQGf1dG9jcnlwdC5vcmeJAc4E  
EwEKADgCGwMFCwkIBwIGFQoJcAsCBbYCAwEChgECF4AWIQRp5NnH84f8yaNXvffHTviz1NECaAUC  
WgG34wAKCRBHTviz1NECaLEtDACaozWd0RHFSA8scsp/J63hHmJplme9eQiei3WTaypWOTwrNXFu  
2evLWETKM2JP5mKHU2EIq551cXxUxUxVOzFZ6/4pJ70Jh8je9sg8/9aZCGLu+0B9VmTKEtOJRitG  
y+AeosZUAKhPhLWvNwCnN3sJfBEuZ8p1febYy1ZqedAlYzr8F6FinBfAiVPXqVbEOCB9dQ2JVgJA  
U/5joG6jDenTOIReerLfC90n3riZs6AFN8LtgDMjx6GJ45WjnpawEYeUd3jqMwZ5HrPkR3CMM3Kx  
XCv8r6Si9UdXQsFonVb2Zi5narwCQpRLi0FwlvretzS0gKdtYOXDWXpgMw7npS0NqCtYf8WG+Nq  
Gq4cwKmlSmYYhvsxIwul34CrA5HNvESUAvPHlenRtfaJehf2KiVx5Wt4tM/RSm+Ls+3U8wKnzGt  
hNcfmAHZ40+45Drmjc/RzWQMGTWLMQd4ueQ79bulcRXrdde+0sRGwEbDLACIU9R8meLXUtWFRN  
jY0/OQS5AY0EWgG3vgEMALLuoUYqjHxHWA5rtGWVgN7s36ypU7KLX8gWk0cz15K0j1K6x+RAQfQ0  
IkcMekC5ShdIt4B7P8+dwWPK75VsTnPk8mlm/QurSNx6KMPbDsr6qy3S6u8C8JicNgutEA3s3nAE  
fkvnMmqz9Z+g8BI5ZUjQAahLW3Qj7g/QKFHhSrYU/7TXGn7o9VvpvKF5HqLdhmpPWI/pUro2sK5  
48R8q8MPXa9fdbE3edYWmAwMP69wYbyC2aW2OnKyI/jfcjiMDabGTSDysCmyJ2F5NU9gLu0ac6qe  
CVCmC8LI2ddQp+h4A8QcVVRsvxYGg25d9Ii5dJVRupKTWWEy7Xak5xEECEZC78y7HAsaA39M9JM  
zsLD/szyrVuo7gXzgyoZJNQ60+b+GrRVJEikrUTddbLaPq/v3hYZaGg3ECuBoY2ISf1OeC7S6Obf  
jwS9WmIG6s5r2IjOaPxx+9xrqh9uUAeEgO9dtDBMfEtW8X6buL9uSXM16z5z6E8L1BqGG4s56OI  
owARAQABiQG2BBgBCgAgFiEEaETzx/OH/Mmjv73xr074s9TRAmgFALOBT74CGwwACgkQR074s9TR  
Amg7YAvcv5Yt3Ja/flXuFhk+TU6WMVz0ehbMIeGw42aW69k78vtEnhWZEYfveE0Vn8Y02/s+n8q  
cimkJfM0TNYnmyb061bCtJG03UsJ84H0zB2L5ws8hTfThy3xqBqaz7hBxki9oK1rIcSeSPfbGa48  
O8w1+FQswFht0L4BTCd/4OfdwLVWFPVgjk/Uzn9vMKxMgtN9+VJ72hwKU/Rf3PnWI6DIKM6MA50a  
YUHxzjYR2KBmq6LJ91rdJ+WUBV7EB2HwtCsx/6kA5gy4ZLQLhrhQz9fS5sjCwFH4mg0i3qTRGxWx  
UwKVvwExHYbqvcEQvWw/13PO6eNjd2qG1Y6uAI0K8Un3UmFeVRQBNmFyX52GqJvMtPdXcawrj081  
Mq1XoBRs6qW+WpX8Uj11mu22c57BTUXJRbRr4TnTuuOQmT0egwFDe3x8vHSFmcf90zG8iKR9ftUE  
+F2ewrzzmm3XY8hy7QeUgBfClZVA6A3rsX4gGawJdo6ZRBbYwckINgGX/vQk6rGs

Autocrypt-Gossip: addr=carol@autocrypt.example; keydata=  
mQGNBFObt8oBDADGqfz6PqW05hUE01dkKm+ixJXnbVripZ2tRkAqT71TF4KBGitxo4IPv9RPIjJR  
UMUo89ddyQfifwKxdFCMDqFDnVRW1DaM+r8sauNJoIFwtTFuvUpkFeCI5gYvneEIIbflr3Xx1pf5  
sLzaERhrHMZMG2farrA+IBympf/BRdceF3rkUu95ssna51/aEEA/YrCFAwCgq7yW700mFlKm/SiCZ  
V4/m0Fae9+Xw+e1WMB+Mav7xL1lvbqGIIPVr0Bzgg8rr4qnJeK/Nx9OvFDD/TepecUfWUUTd8mfYdE  
P/20Js5WGLj1QKUK7LNLdix7deGVhriugVGMsDn5BToj0EXlqi1khOGX2PGz/E+KOWBMnUdu7M1B  
qecfKIIDtwCxBkLd+eRAVf7UPQ+nZV8c/BvDJSGL7Mak3wrd9P2YxmSFditPreemtGHsSE0KdJ7  
Cbg5w4LqD9nTv2CETwFsZep2YABqLe31d1fKESxTJahVTmTWMkBSTaAmtTmbU2tZsr6nJJSAEQEA  
AbQyNDRkYtliODUtZDgWYS00NmQzLWJjMGUtOTM2ZWMxN2Y5YWzkQGF1dG9jcnlwdC5vcmeJAc4E  
EwEKADgCGwMFCwkIBwIGFQoJcAsCBbYCAwEChgECF4AWIQRNY57MDS/rhzDQVtFbQ7jfn25RMgUC  
WgG3/wAKCRDBq7jfn25RMsNkDACS+o9B41TQlTfx+vChiw2Knd1VGCFfSp2TwuiUjKqPLXhwk34N  
pFAbiAUgNNg80Wh9Jkv7b0KX1+eapmkMbNpHWN7u2bMxjerKvp1uuuZNSYGB6YYDDQJrhUariEFm  
eYNMV0r3NGKihfM41+E9rvbrUs4AIWdGn5Wx9mM78XzGy2WSxuI0fRN+zJ5dYphVI+VM8IR5Ah8f  
b/g7c9Vttc1h3ICEuOxJBcvSGafSc+KVj8rSraJnh1vRD/RBwQmWN7Lay0+9GIWGU1xLm5LjWb  
0vZ+9giGbZuzuKutvAu05j6pxdVdNn2yXYKvM6RNTFfrnzRZ0JSuB7Jmcs8xUvJhl1Xk+lLa+x1eS  
r5URQJ7weoqSffKyCojFMd+13tFv3lFhCk+9VYqgAq817ooTeRYH1rXn6vI5qxWjYqpF4Av+iyi  
771j8ES7wRfTsmvHb1ZHYehr20kQr2ibRiv28XdOvXh5UDU4joNvV7btK+uQM0fqqdPQd6RQQ97  
BjJxI/W5AY0EWgG3ygEMAL2AepcoX8fAdRuyaJftxM+9PBiARIu5Nvf8rVbI87+qvENEXoDwft2I  
/iGskpY+5KtnYndfQo8ut/DgHg53Ea6gtDrlz6+FibmmoTXm81tXnTIMf2WhHMq/P343bHRWYqh4  
V3qes1cAzugp8DF8sE4Jte1rCksRjzAWj1BD2X3g2WWSx2wZT4LsIAuDXW0xXSrlVzG8hSK4Xq+4  
6KxaRzG0H+N91X8mj1QMTV16aX5WVI6Rk8WDxpAmBny1CbK1Uu/sRhN40Td3NMQj7bkpcnXeeceO  
/JVbgLEvARG5mj14BwJMSdgXoI12o6/v7xPh+b3cpc7+W6zApvwyhDLaiZjDeS1Sqt/BeEQqiw  
4G75zwbKd3jLRPr/me0/z6J99hQVZYCnIQtfzfxw4dZ6teJ1W7rfkn9BiX1dG94xi2poWR0T4TPjP  
TAFTDSzWdLsPjvklb0/+nhHuGeYzTVEW7Y0yIzZXE5rVmTcuCpFPv+M6VrFfnlCdb5iSjZ341019  
owARAQABiQG2BBgBCgAgFiEETWoeZa0v64cw0FbXwau4359uUTIFAlObt8oCGwwACgkQWau4359u  
UTKYSwv/cALupPaC7kudaEEW1w1Y1KbrKAWPeS0RYIYaPgnrgDFj8e9ThaxMaogD8JRdJ35kZedm  
vInRKvwSCE9NydkJNGBbUkXqInnnuqqg0nFELdWJkfk8+sOhnXDOCrUkAoS6IbUoQ9ua9gFF5kjmj  
+jKhWmNR90k5refGrpp9C61DTxGSOkNqt2Ca7/O6oBovckRNQln75xR04ikvBF9o3VZcfSyDxR+  
eNEb2fMmp66vda8KYncvhBMC3Gi+ablNCfbMP9Lax+pzAB2xb1USBxQcJzDQBmHYLBESAx6IEDne  
c6d9sMH9Y3GPq4aS4M9gFVCCVv+nUGzkYYEordIot2dKTPRQi2Cz//XXrVNg1pXhdtDUgh0mKBuM  
6dyFLBDqDrPcQabyGUJZYhknkQJkt0oASNmHqasjuVhS2N7UuHI+ILMU1SQpBQaCirtTz+CpwKU  
Iy9qsds5eg/4Vvc2AezUv+A6p2DUNHGfMX2FfDus+EP00wgeWbNaV601aE7UhyugB

```
Content-Type: text/plain

Hi Bob and Carol,

I wanted to introduce the two of you to each other.

I hope you are both doing well! You can now both "reply all" here,
and the thread will remain encrypted.

Regards,
Alice
```

## 6.4 Example Copy when a Reply can't be Encrypted

```
The message this is a reply to was sent encrypted, but this reply is
unencrypted because I don't yet know how to encrypt to
``bob@example.com``. If ``bob@example.com`` would reply here, my
future messages in this thread will be encrypted.
```

## 6.5 Example User Interaction for Setup Message Creation

The Setup Code shown in this example can be used with *Example Setup Message* (page 22) below.

```
You'll need to use this Setup Code in your other e-mail app to use
the Autocrypt Setup Message:

1742-0185-6197-
1303-7016-8412-
3581-4441-0597
```

## 6.6 Example User Interaction for Setup Message Receipt

To initiate the import of the Autocrypt Setup Message, the MUA can display a message like the example below:

```
ExampleMail has detected an Autocrypt Setup Message created by one
of the other apps you use to access "alice@autocrypt.example". By
importing the settings from this message, you can start using
Autocrypt here in ExampleMail too!

Please enter the Setup Code displayed by your other e-mail app to
proceed:

      17__ - ____ - ____ -
      ____ - ____ - ____ -
      ____ - ____ - ____ -

[ Cancel ] [ Import Settings ]
```

## 6.7 Example Setup Message

Alice's MUA sends her a Setup Message after showing her a Setup Code (the code used here is the one from *Example User Interaction for Setup Message Creation* (page 22)). The generated message looks like this:

Date: Sun, 05 Nov 2017 08:44:38 GMT  
To: alice@autocrypt.example  
From: alice@autocrypt.example  
Autocrypt-Setup-Message: v1  
Subject: Autocrypt Setup Message  
Content-type: multipart/mixed; boundary="Y6fyGi9SoGeH8WwRaEdC6bbBcYOedDzrQ"

--Y6fyGi9SoGeH8WwRaEdC6bbBcYOedDzrQ  
Content-Type: text/plain

This message contains all information to transfer your Autocrypt settings along with your secret key securely from your original device.

To set up your new device for Autocrypt, please follow the instructions that should be presented by your new device.

You can keep this message and use it as a backup for your secret key. If you want to do this, you should write down the Setup Code and store it securely.

--Y6fyGi9SoGeH8WwRaEdC6bbBcYOedDzrQ  
Content-Type: application/autocrypt-setup  
Content-Disposition: attachment; filename="autocrypt-setup-message.html"

<html><body>

<p>

This is the Autocrypt setup file used to transfer settings and keys between clients. You can decrypt it using the Setup Code presented on your old device, and then import the contained key into your keyring.

</p>

<pre>

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

Passphrase-Format: numeric9x4

Passphrase-Begin: 17

```
wy4ECQMI0jNRBQfVKHVgl+a2Yihd6JAjR9H0kk3oDVeX7nc40i+IjEtonUJt
PQp00tPWASWYuYvJzSuTz9rlyZYV+y4mu9bu9NEQoRlWg2wnbj0UoKk4emFF
FweUj84iI6VWTCsRyMu5d5JS1RfOdX4CG/muLAegyIHezqYOE0Z3b9Ci9rd
DiSgqqN+/LDkUR/vr7L2CSLN5suBP9Hsz75AtaV8DJ2DYDyWYX89yH1CfL1O
WohyrJpdmGJZfdvQX0LI9mzN7MH0W6vUJeCaUpujc+UkLiOM6TDB74rmYF+V
Z7K9BXbaN4V6dyxVZfgpXUoZ1aNpvqPJXuLHJ68umkuIqIyQvzmMj3mFgZ8s
akCt6Cf3o509n2PjvX89vuNnDGJrO5booEqGaBJfWUk0Rwb0gWsm5U0gceUz
dce8KZK15CzX+bNv5OC+8jjjBw7mBHVt+2q8LI+G9fEy9NIREkp5/v2ZRN0G
R6lpZwW+8TkMvJnriQeABqDpxsJVT6ENYAhkPG3AZCr/whGBU3EbDzPexXkz
qt8Pdu5DrazLSftjpjkekrjCh43vHjG18IOiWxKQx0VfBkHJ709CsHmb0r1o
F++fMh0bH1/aewmlg5wd0ixwZp1o79he8Q4kfATZAjvB1xSLyMma+jxW5uu
U3wYUosUmYmzo46/QzizFCUpaTJ4ZQZY1/4sflidsl/XgZ0fD1NCrdkWBNA1
0tQF949pEAE4hSfHfQDNKAY8A7fk81ZblqWPkyu/0x8eV537Q0hs89ZvhSB
V87KEAwXwt60+Eolf8PvvkvB/AK1fWq4MYShgyldwwCfkED3rv2mvTsdqfvW
WvqZNo4eRkJrnv9Be3LaXoFyY6a3z+ObBIkKI+u5azGJYge9704E2DrUEKdQ
cScq5upzXity0E+Yhm964jzBzxnA52S4RoXzkjTxH+AHjQ5+MHQxmRfMd2ly
7skM106weVOR0JgOdkvfiOFDTHZLIVCzVyYV1OUJYYwPhmM1426zbegHNkaM
M2WgvjMp5G+X9qfDWKecntQJTzlyDFZKfd1UrUCPhrv11Ac9cuqgcCXLTdUS
jI+e1Y9fXvgyvHiMX0ztSzlyfvnRt34508G9j68fEQFQR/VIEpULB5/SqKbq
p2f1gJL48kY32hEw2GRPri64Tv3vMPIWa//zvQDhQPmcd3S4TqnTIIKUoTAO
NUo6GS9UAX12fdSFPZINcAkNIaB69+iwGyuJE4FLHKVkgNnNmDwF3f10Czo
hbboWzA3GlpR2Ri6kfe0SocfGR0CHT5ZmqI6es8hWx+RN8hpXcsRxGS0Bmi2
mcJ7fPY+bKastnEeatP+b0XN/eaJAPZPZSF8PuPeQ0Uc735fYlPrrgtWK9Gp
Wq0DPaWV/+0940B/JvWT5wq7d/EEVbTck5FP14gdv3HHpaaQ6/8G89wVMEXA
GUxB8WuvNeHatQ7qXF7TkaZvUpF0rb1aV88uABOOPpsfAyWJo/PEXCZacg8R
```

(continues on next page)

GOQYI6inV5HcGUw06yDSqArHzmONveqjbDBApenearcskv6Uz7q+Bp60GGSA  
lvU3C3RyP/OUclazOp72MIe0+JvP8S5DN9/Ltc/5ZyZHOjLoG+npIXnThYwV  
0kkrlsi/7loCzvhcWOaclvrSaGVCfifkYf+LUFQFrFVbxKLOQ6vTsYZWM0yM  
QsMMYwW5A6CdROT5UB0UKRh/S1cwCwrN5UFTRt2UPDF3wSBACChsHyy90RAL  
Xd4+ZiYf29GIFuwWQyzGBWnXQ2ytU4kg/D5XSqJbJJTYa386UuyQpnFji19R  
uuD0mvEeFvojCKDJWguUNTWsHSg01NXDSrY26Bh1OkMpUrzPfx5r0FQpgDS  
zOdY9SIG+y9MKG+4nwmYnFM6V5NxVL+6XZ7BQTv1LIcIiu+BuJVNWteDnWNZ  
T1UukCGmFd8sNZpCc3wu4o/gLDQxih/545tWMf0dmeUfYhKcjsX9uucMRZHT  
1N0FINw04fdDp2LccL+WCGatFGnkZVPw3asid4d1od9RG9DbNRBJEp/QeNhc  
/peJCPGLGy1A1njTEq+MVB+DHdGNOuy//be3KhedBr6x4VVAzL6jyHu/a7PR  
BWRvt11CIVDxyrEXucHdGQoEm7p+0G2zouOe/oxbPFoEYrjaI+0e/FN3u/Y3  
aG0dlYWbxeHMqTh2F31B/CFALReeGqqN6PwRyePWKaVctZYb6ydf9JV16q1/  
aV9C5rf9eFGqqA+Oix/+XuAG1w0rwlznvtajHzCoUeA4QfbmuOV/t5drWN2N  
Pck2mJlcSmd71x53rnOIgme1hggchjezc4TisL4PvSLxjJ7DxzktD2jv2I/Q  
OlSxTUaXnGfIVedsIOwJfomz5w9tZjC0B5O5TpSRRz6gfpe/OC3kV7qs1YCS  
lJTTxjlmTs6wqt0WjKkN/Ke0Cm5r7NQ79szDNlcc0AVIEOQb3U1R88nNdiVx  
ymKT5Dl+yM6acv53LNX6O5BH+mpP2/pCpi3x+kYFyr4cUsNgVVG1hmkPWctZ  
trHv07wcLrAsrLNqRxt1G3DLjQt9VY+w5qQPJv6s9qd5JBL/qtH5zqIXiX1M  
IWI9LLWHFFXqjk/f6G4LyOeHB9AqccGQ4IztgzTKmYEmFWVIpTO4UN6+E7yQ  
gtcYSIUeJo824ht5rL+ODqmCSAWsWIomEoTPvgn9Qq00YRwAEMpsFtE17k1S  
qjbYyV7Y5A0jpCvqbnGDKZPqCgzjjN/p5VKSNjSdM0vdwBRgpXlyooXg/EGoJ  
ZTZH8nLSuYMMu7AK8c7GMJ1AocTNYHRe9xFV8RzEiIm3zaezxa0r+Fo3nuTX  
UR9DOHOEHadLrFQcfs5y1iRxy9CHg0N2ECaUzr/H7jck9mLZ7v9xisj3QDuv  
i0xQbC4BTxMEBGTK8fOcJHhOABOyhqotOreERqwOV2c100GUQE8QK18zJCUd  
BTmQZ709ttASD7VWK4TraOGczZXkZsKdZko5T6+6EkFy9H+gwENLUG9zk0x9  
2G5zicDr6PDoAGDuoB3B3VA8ertXTX7zEz30N6m+tcAtPWka0owokLy3f0o7  
ZdytBPKly8foTMWKF2vsJ8K4Xdn/57jJ2qFku32xmtiPIoa6s8wINO06AVB0  
0/AutvxcPr+ycE+9wRZHx6JBuJAqQZztU3zu8WZMaqVKb7gnmkWPiL+1XFp  
2+mr0AghScIvJzTDEjigDtLydURJrW01wXjaR0ByBT4z8ZjaNmQAxIPOIRFC  
bd0mviaox61qgQLmSc6mzVlzzNZRCKtSvvGEK5NJ6CB6g2EeFau8+w0Zd+vv  
/iv6Img3pUBgvpMaIsxRXvGZwmo2R0tztJt+CqHRvyTWjQL+CjIAWyoHEdVH  
k7ne/q9zo3iIMsQUO7tVYtGURpRYc2OM1IVQtrgbmbYGEdorhMjaWULg9C7o  
6oDM0EFLCAId3P8ykXQNmluFKlF9il5nr19B/qf/wh6C7DFLOmnjTWDXRiEiP  
6wFEWTeUWLchG1bpiJFEu05MWPiRoRd3BHQvVpzLLgeBdxMVW7D6WCK+KJxI  
W1rOKhhLvVKU3BrFgr12A4uQm+6w1j33Feh68Y0JB7GLDBBGe11QtLCD6kz5  
RzFl+GbgwiPHi3nlCc5yiNwyPq/JRxxU3GRb62YJcsSQBg+CD3Mk5FGiDcuVp  
kZXOCTE2FAnUDigjEs+oH2qkhD4/5CiHkrfFJTzv+wqw+jwxPor2jkZH2akN  
6PssXQYupXJE3NmcyaYT+b5E6qbkiYQj7CknkiqmrqrmxkOQxA+Ab2Vy9zrW  
u0+Wvf+C+SebWTo3qfJZQ3KcASZHa5AGoSHetWzH2fNLIHFULXac/T++1DWE  
nbeNvhXiFmAJ+BRsZj9p6RcnSamk4bjAbX1lg2G3Ssq6MiA1fIRSM1SjuDLrQ  
8xfVFrg7qfBIIQPErJWv2GdAsz76sLxuSXQLKYpFnozVMT7xRs84+iRNWWH9  
SNibbej1h0DcJlKw49Eis/bN22sDQWy4awHuRvvQetk/QCgp54epuqWnbxoE  
XZDgGBBkMc3or+6Cxr3q9x7J/oHLvPb+Q5yVP9fyz6ZiSVWluMeFA9smjJ/A  
KMD84s7u0/8/4yug+swXGrcBjHSddTcy05vm+7X6o9IEZKZb5tz7VqAfEcuk  
QNPUWCmudhzzSNr4+yVXRvpcjsjKtplJcXC5aLuJwq3C5OdysCGqXWjLuU1  
OFSOPvTsYC2VxYdFUcczeHEFTxXoXz3I0TyLPyxUNsJiKpUGt/SXmV/IyAx+  
h6pZ2OUXspC9d78DdiHZtItPjEGiIb678ZyMxWPE59XQd/ad92mlPHU8InXD  
yTq6otZ7LwAOLGbDR9bqN7oX8PCHRuu30hk2b4+WkZn/WLd2KCPddQswZJg  
Qgi5ajUaFhZvx5YNTqIzzYVh7Y8fFMfzH9AO+SJqy+0ECX0GwtHHeVsXYNb  
P/NO/ma4MI8301JyipPmdtZvvt9NOD/PJcnZH2KmDquARXMO/vKbn3rNUXog  
pTFqqyNTr4L5FK86QPEoE4hDy9ItHG1EuiNVD+5suGVGUgYfV7AvZU46EeqO  
rffj8wNSX1aK/pIwWmh1EkygPxsomWRUANLX1j06zX9wk2X80Xn9q/8jot1k  
V1540od7cvGls2wKkEZi5h3p6KKZHJ+WIDBQupeJbuma1GK8wAiwjDH59Y0X  
wXHAK7XA+4tu0dgRpZBUUMqQmvEvfJaCr4qM1puGdEYbbpIMUB1qCFYU9taL  
zbepMIT+XYD5mTyttZhr+zrsfpt1EzbrhuabqPioySoIS/1+bWfXvndq16r0  
AdNxR5LiVSVh8QJr3B/HJhVghgSVrrynniG3E94abNWL/GNPS/dTHSf8ass  
vbv7+uznAdzHsMiG/ZlLAekQJ9j0ENJvHmnyeVFIXDV6jPccQJ+rURDg17z  
/qTLfe3o3zBMG78LcB+xDNXTQRK5Z0LX7h17hLSElpiUghFa9nviCsT0nkcr  
nz302P4IOFwJuYMMCEfw+ywTn+CHpKjLHWkZSZ4q6LzNTbbgXZn/vh7njNf0  
QHaHmaMNxnDhUw/B113uM52qtsfEYK07SEhLF1JbAk0G7q+OabK8dJxCRwS3  
X9k4juzLUYhX8XBovg9G3YEVckb6iM8/LF/yvNXbUsPrdhYU91PA63xD0Pgb



```

zthZCLlInF+lS6e4lWjv3n1dc4dFWD7F5tmt/7uwLC6oUGYsccSzY+bUkYhL
dp7tlQRd5AG/Xz8XilORk8cUjvi6uZss5LyQpKvGSU+77C8ZV/oS62Bds5TE
osBTrO2/9FGzQtHT+8DJSTPPGR6rcQUWLPemiG09ACKfRQ/g3b9Qj0upOcKL
6dti0lq7Aorc39vV18DPMFBOwzchUEB1BFyuSa4Aod30tsoilAC3qzbWbu3z
QLjms76HEcWDkxgDAh1Bz6/XgiVZsCivn7ygiGmc2+hNEzIdDsKKfM9bkoe
3uJzmsv8Bh5ZEt fGoGNmu/zA7tgvTOCBeotYeHr2O6pLmYb3hK+E/qCB114
8pK4qYr jAlF+ZMq9BzXcaz5mRfKVfAQtghHOaNqopBczSE1bjFF6HaNhIaGa
N8YdabNQG7mLI/fgBxJfkPl6HdIhEpctp4RURbSFhW+wn0o85VyHM6a+6Vgj
NrYmhxPZ6N1KN0Qy76aNiw7nAToRRcOv87uZnkDIeVH8mP/Ohldiy/Y97cG
QgOeQHOG27QW57nHhqLRqvf0zzQZekuXWfbqa jpaabEcdGXyiUpJ8/ZopBPM
AJwfyA2LkV946IA4JV6sPnu9pYzPzXQ4vdQKJ6DoDUyRTQmgmfSFGt fHAozY
V9k0iQeetSkYYtOagTrg3t92v7M00o/NJW/rKX4 j j2djd8wtBovOcv4kxg4Z
o58Iv94ROim48XfyesvSYKN1xqgbXH4sfE6b4b9pLUxQVomWANLK9MK8D+Ci
IvrGbz5U5bZP6v1Nbe9bYz jvWTP jaMr jXknRTBcIkavqOfDTSIVftT4qvhvK
42PpOrm0qdiLwExGKQ9FfEfYZRgEcYRGg7rH3oNz6ZNOEXppF3tCl9yVO1Fb
ygdIeT3Z3HeOQbAsi8jK7o16DSXL7ZOpFq9Bv9yzusrF7Eht/fSEpAVUO3D1
IuqjZcsQRhMt IvnFOoFu jFtoOjx9x3dj/RarvEGX/NzwATZkgJ+yWs2etruA
EzMQqED4 j7Lb790zEWnt +nuHdCd1PnNy8RG5u5X62p3h5KqUbg9HfmIuuESi
hwr6dKsVQGc5XUB5KTt0dt jWlK5iaetDsZFuF5+aE0Xa6PmiQ2e7ZPFyxXmO
T/PSHzobx0qClKCu+tSWA1HDSL08IeoGZEyyhoaxyn5D9r1Mqq101v/iu59r
1RRs+plAhbuq5aQA3WktF1N6Zb5+AVRpNUyrxyHoH36ddR4/n7l1ld3STGD
RqZlRouKHS3dcNW2Pt151U+loYsWFZwC6T/tAbvwhax+XaBmiKQSDfMg9sBw
TiM1JWXhq2IsjXBvCl6k2AKWLQOvc/Hin+oYs4d7M9mi0vdoEOAMadU/+Pqn
uZzP941mOUV5UeTCBbjpyfI7qtIi3TH1cQmC2kG2HrvQYuM6Momp//JusH1+
9eHgFo25HbitcKJ1sAqxsniYIw5/jIVyIJC7tatxmNfFQQ/LUb2cT+Jowwsf4
bbPinA9S6aQFy9k3vk07V2ouYl+cpMMXmNAURboFRLxw7QDapWYMKdmnbU5O
HZuDz3iyrm0lMPsRtt/f5WUhZYY4vXT5/dj+8P6Pr5fdc4S84i5qEzf7bX/I
Sc6fp1SdYBscfHdv6uXsEVtVPKEuQVYwhyc4kkwVKjZBaqsGjAA7VEhQXzO3
rC7di4UhabWQCQTG1GYZyrj4bm6dg/32uVxMoLS5kuSpi3nMz5JmQahLqRxh
argg13K2/MJ7w2AI23gCvO5bEmD1ZXIi1aGYdZfu7+KqrTumYxj0KgIesgU0
6ekmPh4Zu51IyKopa89nfQVj3uKbwr9LLHegfzeMhvI5WQWghKcNcXEvJwSA
vEik5aXm2qSKXT+i jXBy5MuNeICoGaQ5WA00J300h5dn0XpLtFUWHZKThJvR
mngmlQCMMw2v/j8=
=9sJE
-----END PGP MESSAGE-----
</pre></body></html>
--Y6fyGi9SoGeH8WwRaEdC6bbBcYOedDzrQ--

```

When decrypted with the Setup Code, the encrypted blob at the end contains:

```

-----BEGIN PGP PRIVATE KEY BLOCK-----
Autocrypt-Prefer-Encrypt: mutual

lQVYBFn+zZUBDADBo2D+WUbm3lN1lXtQTxLhxVADIIMLK1dFUgu5w1KAMrW0x9x2
7cRNxzVrTfiv2FiwThUHZmJBFai8HtsMvn/svrCPEgPvkjTDMCWZaEEc5/g51Uys
zjf6fUsGXsC9tUcva6pGHaTe8Iwpz5stKjRKI3U/mPdQpXmaurwzEdv1NWNi9Ao2
rwWV+BK3J/98gBRFT8W6gv+T/YGXVrqXMoMMKLTfze2uy00ExJkhI64upJzD0HUb
GjElYdeSWz71YhQ2y5cmnWPfrnOxiOCVyKrgBulksda5SIjEqCJCVYprX/Wvh5fe
RXYftWVQUMeo6moNoHTM9X+zQJPWWuWivOJpamIuUCziEycX8RtRo0yAOPwc/vIp
poxAMusQCVn15YwVEcngzXUi3EB72wXJ4411VfzPCSlgVNzV7Yqx1lW4PMRcFB2o
blO25rk3GDlmgEVCg1Hh4FtEBkmwVjiv4duNOE33r2Yf8OsFAkKnRCR11Yn8409D
aJGou41hEV+LAsUAEQEAQAL/i2DNOQ7gCR565RmzMvYtheuPIrrnJlmt7WxndNs
8wpyQM6rrige5QWh9a6RrkrIdzoDNEKfwCbLjDQhLXu+18tBm7axBY4052VcPu4i
eLFuXWPcfe/ejX447kYiRbuhLMjazbP6u jpzQAKAyxiPw6gMUv3eenyVbD33g3D
3BMw2/oRYYguVYoe+4MkqdJtuTX8VL1s111G16vGRQeOJggY07ptVzj+fWUiP1qw
a/uHEdidebtj0FrYtYt f6hDB5QNKr6X3Bax+1N82mJI4iGCONbWpZqCty+LXub6
Q9B5V5gB6P9A3RfwpgeJ0H8y/WfgT9Jfmzq+fWmtAdvftkHA941lbYwFuUxIk1f
HqESWo311LxG59PxxvBtRWRVACW2Hzz7IcAmhEJAZkEUbGkn5o1qKBrNjX9/4nG
wKfVfXc358KwvRd64pZnZrjv7f7CEhFIcWNeWyFjaG0Cq1isGxanxzUcH+SO1gHx
w7b6e5S1+G19+b1FRITt+wk4yQYA16SgrvPzXj3Mat238BsosX5N+6RL760HjXoU
SC1E0UAgFvXVouWuGMSA/p41nDkwN8dPkVP+8AXYc0mgsCv/5jOgm9Px1uI2LUGEa

```

```

0ZLN3+XFcpxxvEILcfErrwlPPL8lmg5cK2NHNCSpwbEUssiLd11uQO3IzEFrfc0
GMARweu4Vr9pbD5Qrvaea+TATeO1Hj2dDE0EJJDEduWiKWhNKG6wp3z4MhGpuUN/
CSywaZiy4V3HapPt5t0ckAVVTaYJBgDl4IGLXHjrEke7aplWHulzSxjtPupyVLBj
RjhVhKZUtPu11ETg3SxW0cdyAy1iCt6rs4Hpp19HYcJE3mWYDfn+B8R3+HGHOHhs
uynnLzx5WD4xsWVFAEluvVjzWcOnQnxamUzHfE+5+8GuTechZjGrPVvZddMg09DV
5QU6tqOUfie3tmJu5KSEdfzIomL7p3ZNcEcLr6tSdyHq6Xa1Ft27Y6xNdWdad1I
KO+FamsTlGULQnpINwj4Ee7ZVJAhd0F/iOFZh4c5nmox8asjOB9wyEvzEu3i1W/
Rh3EDTMLKjWfz3H8LFxc/vt+T8LDn9paggV4K5OH8v211lhYlUezygVFRRXhtbt1
pvoN/sAnZsvii0PXec8vM7kttX583LyFOphuMFZOrAii47VvYUqzBTrKdggwxdjE
NagvKTQhsGIJWh50jHR0npOHazDKZcwfYvNzPuRiYURsIxXeYak3i3d2Lg6acxA
wnySqvFKOVsQlROYzxbUspVi3X6YBIpwXOSXtDjHMWViZDY4ZC04Yzc3LTQ1Yjgt
YjAzMy04Y2FjM2Y3ZDIwNmRAYXV0b2NyeXB0Lm9yZ4kBgZTAQgAOBYhBOYEAm5E
13w/zp/QcnHbxWV/3mWnBQJZ/s81AhsDBQsJCAcCBhUICQoLAgQWAgMBAh4BAheA
AAoJEHHbxWV/3mWncL8L/222EH1DqjLKMRE9mZFjdXyfrTB3SHfm4upB9xvnVRgp
neP7rWdyTPaIH0utHFj1DfVajMyrNr4nN7j+D9VgcuVlTmDQzeksRntITIO91VPn
bcFUWwJDCOSrrv0kZn/E/Mk49pw51cWwo8R82/MqAr7HRrhDxvTdj6YvmaYY8Gu
e4LNr+cWF69StBtu25TOEGcwUgW8q/NZRMocSAgMurP7xq485B1JsXYP/UES+1uh
t2BCL5gktqPvv+1RFHWSnuY7nUh99OzSqAwmmHyPBbiUxAYgJPLjd6pPXL1AT4Mf
1EEBilxEKZNWetLnxqmdakf9rF8IONuhbAPraA3R1rBztYRD6t2C7xZOhyjJgDqL
IKTpezn2Y4YTSCwJ1m/Mqu4k5iq8RHN40JsNzeFcoM4TzaiQGNCGw5UIrdru7IAh
mdzP0qi+LQKRd13cS4bzb3sdJ/X8+6myIWAcGwnOZnnj35kcteVnmyzhqP0e17ts
KTyhrQv4DrX6clhWxUNI7Z0FWARZ/s81AQwA0jF8OQSOCGRKcxvOodpQCIGH4ZI
xdQPt1CfbxkbFH/ZjnC7s7kx2Q8woiuzJCjBJ4afXyuczU/GdEY6tf5CdV1N2Tvd
V4wgPqczVNN+/mCaSNxvo2mEY945NnIkhuOBDETPYtRuEUux5FL/oI4XmrpOP5Mk
VI9sOzmRWbwuoCtra9292nFXrlY/YV/PACgppPETCcMpzeunvIQjnarPzExMI74i
QEhz2vB2PtOonEw5N1B1+lJ+W2IbCDeUIZhoe56MnMNCVT9fo4ISr9ZPv9RWo3Bm
SuxPi4b0EUZb5Y5e46mAdi/RhDrZdACi1U1dRjXRCwTJvoNovq9iN3QuT/PuJwBf
m7OOV8k3dNwonFLSKNa19gPnYH3fr6aLMZH73u7KoQFU1ArDDWm8p1kOu6JHjc7S
TrdMw7/hwCFd/Dur3X9EwaBmlfZQL8EYyJ4/OJug/4YdfzuFGYC8UJGNBzQoXLEk
Zs0ogPcqf9GFst48IBVYjfiVJDQJmouVGf1ABEBAAEAC/4tr+ez76K7vf8fQ0r4
NjJAdJ4zr0BVKGGzBkVkrJ1PUvryG1ub84mbI1NAR42TM/1IrRgpe6XENEyN/C5p
28TPUrWZ2wofqW9d9oIwMxf0SoP1h10H75iLiOI3zEZWf47OHw1QbhkuzpvuosA2
QXNtWATGCEfZNGOCGqCV11Gt00nxIzvOBbiZvX2gWM15Vmp+X3Y/w6w14D4tmI0
M8meHc3lbb7taCGYvVd1j5QjReigPovpeRpsu21jE4sw4vma/IZuiEgO+0JPA58K
atGP+y1mEHT78KyKc7EdJY+Pw9a4uD2eTdnOiHjOdFyBVf/JHX/nG0dBQrnL14J9
lQbGGQXxl13qo5v9jP6NZJ+IC4/ONYmLBFFS5QJW4rWveCO49wDjuPh5HVO4yvrX
KrxVA8GckbV9ho3gCbJyMoqfNdcEtzbzGkzc84W+alVrUUKbuUEPK6j+auGTL1PII
Wym6hqHPEN0bkr3qolwn6nCyYz2J83RqgMKmw5Ovcz5zmjEGANR2GBQs0rYY5m3z
x2ISPu1ZHpaJW7UB1RfgmhCQ78N1UPOji8Qp2/Ehj94+/OULmTUKCTNXeFlt0PzF
atiOQWohM8aoA7K6ZJrk+PdPTu6/2seEtPm6YfaIMGO9TJgxc15hC6jDc7x4wxj9
1Bw9zVzFGpRtFsgawVh0+BoM2tQ17R4oWVjXopGRUkznB/ZJiZXDbxeq71NcqQou
6uib2SF3aMzes/a+CdQR6GC+cGNAEz3YRb6d4dsEmp3xQrEsRQYA/Uw95K8jjiYs
GSngKdpfAE8rEbn6Au92OKONEE1OvdFFuLg+m8R2TYXr9U8j5bA961vKvSe/nAUj
jn7Vjnk30Fo05htW0agkGIKUUDFS6Z1jGdJWRD67IM+GHLHoVkiSDCY0JLS76H07
JC/P08j+2K6I0sYqx8TUTywMPGtIRDE11gJwPTXKnV9H7WTbqqjNgWR3dalKKLY1
Ox76ZMCjn6JrkYR1WnHkIjLZSVLnPMSeohm7KvYwrnma4rvGpf/xBf9QvfZAJF8J
2Ez6LFePDA8joX9m75yXh1ClfPjPmhu4+gaaNPU7+S8gU52BvD6AFqzJQsVwZmB9
uzqiKQooqez1Js9zP/6+sPk91SmZzdVljQ4/JwaiCptw9/tGW8/nFQxNeg0jd0JV
IFPmop0+ouvyTINKfn69AgU3BuBGo+kTXRbjV7Q7JNdFFjSKBK56ptFJvR/h4mpE
0Lxvl0gKnmDxWYyE0Byquak0hd75O209ttrWeatE1b1o4bV0+A1Osi71xIkBtgQY
AQgAIBYhBOYEAm5E13w/zp/QcnHbxWV/3mWnBQJZ/s81AhsMAAoJEHHbxWV/3mWn
miML/1kdi2CpT13v9bDCn4fokmHiY76sdeYuDmi7pqJ7fm7WZqcmA1PLDmjAddqA
YEN7DWGkKX5E5P0Dcn5W7okTjyXgDUMwuwpI90gwRaDF8qsZp84R9D9ar0/dFTgd
OtT9Wh407rLlOPjLryyq4L2i7cyuMbohyM6ZEwr7XMjZokuUIToLj1d91EOh3HEi
BGmTucPs+mv1dCdWdfZVcDpzmrvKeA7Ax60Cn3FCqTVCqfBoJDoSz+w5rKnZZ0KCG
sOD8Z0rIOx+YphyhdV6P/J4dBUvpeZKSXp3YiNWRsv8hEozfYtZCkqi+f/keD5E/
X6AKKLaCt06y23Mh7sRY+bpnFLqqhn7L44YAv2SMr76EX+F9AZ59YfYaaOmbwaDw
zOZScbVC+uGceR1y3egkFxn2X2VXjPjg6kMiExkE/qe7jA4mReNgyok8iYyRwAYI
lfideiDOMKGhnsAFPtFYPiQ7n+xHPiiseVDQyNfDyU08xlaeuRr89jKvwh0/6Xh
TRzalg==
=f96/

```

(continued from previous page)

-----END PGP PRIVATE KEY BLOCK-----