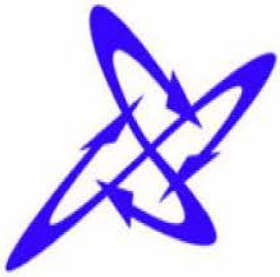


Using DomainKeys Identified Mail (DKIM)

LISTSERV®

Version **17.0**

L-Soft

Using DomainKeys Identifed Mail (DKIM) with LISTSERV®

25 Nov 2019

This feature is not available in LISTSERV Lite.

Contents

Introduction and Prerequisites

In order for DKIM support to work, we assume that DKIM support has already been configured in DNS for the domains you will be signing for, per the DomainKeys Identified Mail documentation. If not, general instructions are provided below.

DKIM support is available for LISTSERV Classic and HPO, on all operating systems except for IBM z/VM. It is not available in LISTSERV Lite.

Important: Support for DKIM replaced support for the old Yahoo DomainKeys system, which is now deprecated, beginning with LISTSERV 16.0-2017a.

Most LISTSERV sites already using DomainKeys authentication will find this to be a transparent change when upgrading from an earlier version, with no need to make any adjustment to your current settings. However, see below.

Important: Sites running older versions of LISTSERV which supported the Yahoo DomainKeys specification will wish to review their existing key pair before upgrading to LISTSERV 16.0-2017a or later, as key lengths which were sufficient for DomainKeys may be too short for DKIM. Per RFC 6376 "DomainKeys Identified Mail (DKIM) Signatures", Section 3.3.3, "Signers MUST use RSA keys of at least 1024 bits for long-lived keys", whereas many DomainKeys sites *may* be using keys of 512 or 768 bits.

In addition, RFC 8301 updates RFC 6376 and states in Section 3.1 that "DKIM supports multiple digital signature algorithms. Two algorithms are defined by this specification at this time: rsa-sha1 and rsa-sha256. Signers MUST sign using rsa-sha256. Verifiers MUST be able to verify using rsa-sha256. rsa-sha1 MUST NOT be used for signing or verifying."

Bottom line: L-Soft strongly recommends that all DKIM keys MUST be 1024 bits or more, and they MUST be SHA-256 (AKA SHA-2) keys. Keys created with the SHA-1 algorithm are NOT supported by the DKIM specification, and MUST NOT be used. *Key pairs NOT meeting these specifications are used strictly at your own risk.*

Creating DKIM Keys and Configuring DNS

Information describing the creation of DKIM keys and the configuration of DNS to enable DKIM signing is found at the DKIM website. Please see http://www.dkim.org/#specifications for the official DKIM documentation.

Creating a DKIM Key Pair

It is quite simple to create a DKIM key pair. There are websites where you can enter the basic information (selector and domain name) and the website will generate the key pair for you. However, it is questionable whether such sites will actually guarantee the confidentiality of the public keys they generate, so this may or may not be the best route for your site.

Otherwise, the simplest way to generate a DKIM key pair is to log into a unix machine that has OpenSSL installed, and issue the following commands in a terminal window:

```
$ openssl genrsa -out rsa.private 1024

$ openssl rsa -in rsa.private -out rsa.public -pubout -outform PEM
```

This should result in two files being created: rsa.private and rsa.public. The rsa.private file contains your private key, which will be used below to create the DKIM file for LISTSERV; the rsa.public file contains the corresponding public key, which will be used to create the DNS TXT record you need for DKIM.

Creating DNS records for DKIM

Many of our customers have hosted DNS, that is, the domain registrar from which they have purchased their corporate domain(s) also hosts their DNS zones, and these customers typically edit their zone file via a web-based GUI. In that case, for this and for each of the following examples, simply enter the appropriate information in the GUI and follow the registrar's instructions to save and propagate it.

Other customers (generally large corporations or academic institutions) are more likely to run their own DNS servers, and will have to edit the appropriate zone file in the usual way.

In either case, this section is intended only to provide examples of the information you will need to create your DKIM records, and general DNS advice applicable to both cases above. Editing zone files is beyond the scope of this document and customers in the first case, above, should consult their ISP's support for assistance, whereas customers in the second case are urged to consult the DNS/BIND documentation for their particular implementation of DNS for guidance.

Important: Please be aware that the examples provided below are not intended to be used "as-is"; you must substitute the correct information for your site or DKIM will not work. In particular, please note that the public and private keys in these examples are purposefully invalid and cannot be used to create live DKIM records.

Creating a DKIM TXT record in DNS

Creating a DKIM TXT record can be done in various ways. If you run your own DNS, simply edit your forward zone file to include a TXT record. We will assume for this exercise that the LISTSERV host name is "**listserv.example.com**", and we will enter the following information:

Host:	`default._domainkey.listserv`
TXT Value:	`v=DKIM1;k=rsa;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQ KBgQDcARWuStG7G33L+M5jqjiCbhfKBlgxIMC8Of5ODONOTUSE THISKEYITISANEXAMPLEONLY91O1RigBB/C+UXzPO+N1+hZ55Z XS8MPGPgaV9VM1EysEdyfm2Y/rn935GGJwtm67fz+6dyKkCAzL sMjR5DvcxxlMzf6Gs9TrX7PBNwIDAQAB`
TTL:	Your preference, but typically 1 hour

Notes:

1. When creating a new record in a DNS zone file, the host name normally is not fully-qualified. If you are editing the zone file for the example.com zone, it should not be necessary to enter the fully-qualified domain name in the "host" section. *Be sure to*

check the documentation for whatever DNS you are running if you are unsure of this.

2. We are assuming a DKIM selector value of "**default**". For the purposes of DKIM authentication, external sites will always check DNS for a TXT record belonging to "selector"._domainkey."hostname". In our example, external sites would be looking for the TXT record belonging to **default._domainkey.listserv.example.com** .

3. The TXT value should not break and wrap as displayed above. It should be one continuous line of text. The value of "**p=**" is the text of the public key from between the lines

   `-----BEGIN RSA PUBLIC KEY-----`

   and

   `-----END RSA PUBLIC KEY-----`

   Those lines should not be included as they are not part of the public key.

Creating a DKIM TXT policy record in DNS

This record is optional, but recommended. Newer versions of the DKIM specification may actually require it, so it makes sense to go ahead and create it while you are creating everything else. Again using our "**listserv.example.com**" example, you will enter the following information:

Host:	`_domainkey.listserv`
TXT Value:	`o=~`
TTL:	Your preference, but typically 1 hour

The "Host:" field is slightly different this time. The policy record does not require the selector "default", so we leave it off. **Note that the underscore before "domainkey" is required.**

The "TXT Value:" field contains the policy to be applied to DKIM lookups. The value we've provided above means that "some" outbound mail from (in the example case) listserv.example.com will be signed with DKIM. This is the default, and L-Soft's recommended setting.

LISTSERV Configuration

LISTSERV's DKIM support is configured by doing two things.

1. Supply one or more private keys.

   Each private key is stored as a text file in LISTSERV's main or home directory (that is, the directory where the *.list files are) and must be named *xxx*.dkim, where *xxx* is the arbitrary name you choose to give the key. If you only use one key, it is recommended to name it default.dkim.

   The file is created in the usual openssl/RSA format, with one minor modification. Here is an example:

```
d=listserv.example.com; s=default
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDcARWuStG7G33L+M5jqjiCbhfKBlgxIMC8Of5OQaM00v83IRuk
jSq4pPvAhsHKSCacVCHp91O1RigBB/C+UXzPO+N1+hZ55ZXS8MPGPgaV9VM1EysE
dyfm2Y/rn935GGJwtm67fz+6dyKkCAzLsMjR5DvcxxlMzf6Gs9TrX7PBNwIDAQAB
AoGBALY1V8WARe+XNzqlmBnHMwIjOCSj2Irnu3io90vM5OStE56PFxvTptxCGBc+
BGYKF6BFtcjWhEeQETW5Y9PcHWbj3O2OSrhk9sPQHZCW46J0IVpP0vRHyrK4o+zX
CbHkFEJZFSBN2IquUR5m9Yqb5dqQPRf/7lGAQpVrd03wiX4RAkEA8jRE3CFfh7I5
idx1q2ohBEh2rPHioDONOTUSETHISKEYITISANEXAMPLEONLYPhcwoDjQQ/EqIUS
wezkWNX2zQJBAOiJGr7tzHY2Cg4ftfl1DJYXNkRtsR4ZoVsgcjhPVTLScfG7nOFL
pMCKE5ChYFkbYmh5knhOsYrZgBqPDxe8MBMCQFY3dv+pPZlPPx4tBRIUwFYG+X/M
xvGpwDhMaYIm5fmlwBLCBnHt8Z+kEGVwKbabVUkcLHUmYjOe0zOHAS4CVE0CQHSA
9MCCHfV//6ux4Zd5OHQebxb7qki9aKVibTefL72FyIbni6MpJgM9aq4E3GPon3Ze
qq7SJou9izxDPrmSlLcCQBG0OYhOQWank6kWaziTY/K93vGyHQOqUM425iLQdWWu
DHj08akKRILiTXhUYgQA9/fE/ncalK4ChvsVG0bqXZ0=
-----END RSA PRIVATE KEY-----
```

   The first line in the file must include a specification for the 'd=' and 's=' parameters of the DomainKeys signature (in whatever order, as long as they are both there). Per the DKIM documentation, these variables specify the domain for which you are signing ("d=") and the "selector" that is used to form the query for the public key ("s="). For instance, let's say that your public key is registered as follows in the DNS:

```
default._domainkey.listserv.example.com IN TXT "g=; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDcARWuStG7G33L+M5jqjiCbhfK
BlgxIMC8Of5ODONOTUSETHISKEYITISANEXAMPLEONLY91O1RigBB/C+UXzPO+N1
+hZ55ZXS8MPGPgaV9VM1EysEdyfm2Y/rn935GGJwtm67fz+6dyKkCAzLsMjR5Dvc
xxlMzf6Gs9TrX7PBNwIDAQAB"
```

   The selector is "**default**" and the domain is "**listserv.example.com**".

Important: Again, please remember that the public and private keys in these examples are purposefully invalid and cannot be used to create a live DKIM configuration for LISTSERV.

2. Supply a DKIM_SIGN Configuration Variable

   In your site configuration file, add a DKIM_SIGN= variable containing a blank-separated list of domains that you are able and willing to sign for. You can use wildcards, but only of the form '*.EXAMPLE.COM'. You can't use, for instance, 'SALES.EXAMPLE.*'. For each entry in the list, specify the key to be used, as follows:

```
DKIM_SIGN=EXAMPLE.COM *.EXAMPLE.COM EXAMPLE.CA(CA) *.EXAMPLE.CA(CA)
```

In the example we have been using above, our DKIM_SIGN variable would be

```
DKIM_SIGN=LISTSERV.EXAMPLE.COM
```

(Under unix, don't forget to export DKIM_SIGN .)

By default, the key called DEFAULT is used (if one exists). So, in the sample above, the key for EXAMPLE.COM will be fetched from DEFAULT.DKIM whereas the key for EXAMPLE.CA will come out of CA.DKIM.

Starting LISTSERV with DKIM Support

LISTSERV loads the keys at startup and makes simple verifications.

```
26 Jul 2019 14:14:26 Loading DomainKeys private keys...
26 Jul 2019 14:14:26 -> Loaded DEFAULT (d=EXAMPLE.COM; s=DEFAULT; RSA-1024)
26 Jul 2019 14:14:26 -> Loaded CA (d=EXAMPLE.CA; s=DEFAULT; RSA-1024)
26 Jul 2019 14:14:26 DKIM support enabled
26 Jul 2019 14:14:26 DKIM Accelerator enabled
```

In particular, the 'd=' parameter in the key must match or be a parent of the domain you want to sign for. Thus, the key for EXAMPLE.COM can be used to sign for EXAMPLE.COM and *.EXAMPLE.COM, but not for EXAMPLE.CA. LISTSERV will skip any invalid entries. Keys are kept in memory so you can have as many as you want (within reason).

If there is no DKIM_SIGN variable or if you are running a LISTSERV version without DKIM support, LISTSERV does not attempt to load any keys and the DKIM feature is bypassed.

Using DKIM with LISTSERV

By default (`DKIM_SIGN_ALL=0`), LISTSERV does not sign any messages using DKIM other than those for which DKIM signing is expicitly requested by the caller, for instance, DISTRIBUTE jobs with an explicit "`DKIM=YES`" parameter in the JOB card. List mail and non-list administrative messages will not be signed when DKIM_SIGN_ALL is left at the default value.

However, because of its relationship to the DMARC protocol, you will probably want to have LISTSERV sign every message that it generates, regardless of its source. Setting `DKIM_SIGN_ALL=1` in the site configuration file tells LISTSERV to try to sign every message for which it has a suitable private key, as defined in the DKIM_SIGN configuration parameter (see above).

(If setting DKIM_SIGN_ALL in the go.user file under Unix, please also ensure that the variable is exported.)

Once you have enabled DKIM signing with `DKIM_SIGN_ALL=1`, the behavior is as follows:

With mailing lists:

- Incoming DomainKeys signatures submitted to a mailing list will be suppressed unless "`Misc-Options= KEEP_DKIM_SIGNATURE`" is set in the list configuration.

  In general, you will not need (or want) to use the KEEP_DKIM_SIGNATURE option. As DKIM is specified today, signatures DO NOT survive posting to mailing lists (LISTSERV or otherwise), so LISTSERV removes them by default to avoid triggering alerts for subscribers whose mail hosts have implemented the stricter forms of DKIM.

Therefore, if used at all, the KEEP_DKIM_SIGNATURE option should be used judiciously and with caution.

- When DKIM signing is enabled at the server level (`DKIM_SIGN_ALL=1`), the default is that all list mail (including administrative mail) will be signed. It is possible to override the default and disable DKIM signing for individual lists (typically for debugging purposes) by using the "`Misc-Options= NO_DKIM_SIGNATURE`" setting in the list configuration. *It is not recommended to run with this option set during normal operation.*

In DISTRIBUTE and DISTRIBUTE MAIL-MERGE jobs:

A `DKIM=NO|YES` option is available for the DISTRIBUTE command (default: NO). This will fail if running a LISTSERV version without DKIM support, but otherwise it always succeeds. Messages originating from domains for which LISTSERV has been configured to sign will be signed, while those originating from other domains won't be.

In other types of messages:

When DKIM signing is enabled as described above, LISTSERV will to try to sign every message for which it has a suitable private key, as defined in the DKIM_SIGN configuration parameter.

Restrictions and Implementation Choices

If DKIM signing is enabled (`DKIM_SIGN_ALL=1`), a message that already has a DKIM signature when it arrives at LISTSERV will have that signature replaced by one generated by LISTSERV.

LISTSERV can be configured to retain the old signature of such messages via the list -level keyword setting "`Misc-Options= KEEP_DKIM_SIGNATURE`", though (as noted above) this is rarely, if ever, recommended. With this setting, only the original signature will be included in the distributed message. LISTSERV won't add its own signature in this case, as double DKIM signatures are disallowed in most cases and, even when allowed, may not be handled correctly by all implementations.

DKIM can be used to sign mail-merge messages, but in that case LISTSERV's Embedded Mail Merge (EMM) feature MUST be enabled. Using EMM is the only way to guarantee that the signing engine will see the exact text being sent to the recipient, and that the signature will match. **EMM is normally enabled by default,** but can be disabled at the server level. Check the setting of EMBEDDED_MAIL_MERGE= in the site configuration if you believe this may be an issue.

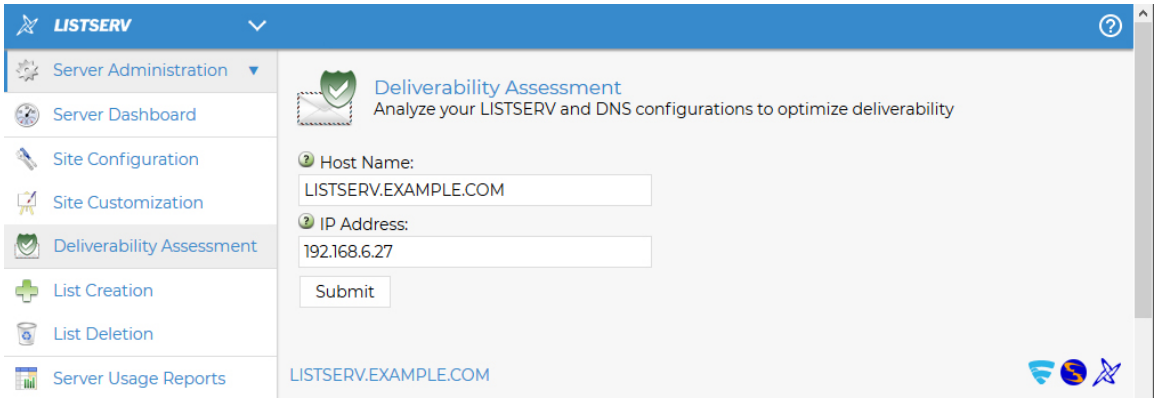
Testing DKIM

Once you have created your DNS entries and LISTSERV configuration for DKIM, you will want to test it.

LISTSERV's Deliverability Assessment report

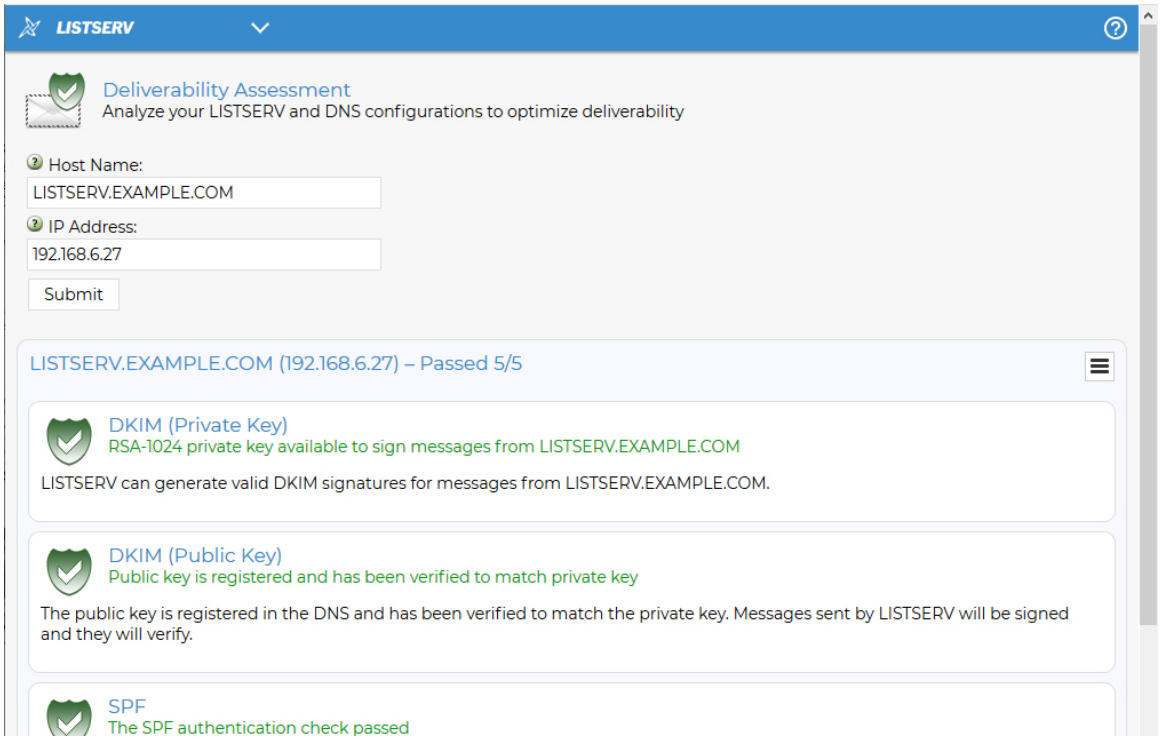
LISTSERV has a built-in Deliverability Assessment report which can be reached in the web interface at **Server Administration** -> **Site Configuration** -> **Deliverability Assessment**. The first screen looks like the following:

Figure 1 Deliverability Assessment - Initial Screen



Unless you have multiple domains set up in LISTSERV, there should be no reason to change the pre-populated values. If they are correct, simply click **Submit**. This will yield the report:

Figure 2 Deliverability Assessment - Report Screen



The green shields indicate that, so far as LISTSERV is concerned, you have properly configured the DKIM DNS entry, and LISTSERV itself is properly configured to sign outbound messages with DKIM. If either or both of the shields are not green, you need to recheck your DNS entry and LISTSERV configuration, and correct any errors before running the report again.

Testing the DNS entries

Once you have created the DKIM DNS entries, you can check them with NSLOOKUP or DIG to ensure that they are being served properly by the DNS server:

If you use NSLOOKUP to check the record after you create it, you'll see something like this:

```
> set type=txt
> default._domainkey.listserv.example.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
default._domainkey.listserv.example.com       text =

        "v=DKIM1; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDwl88WzSt
oOTznXCeMJF+J0XDaxrRYgl42hx+zZ4MUAaI8ZCsaozbK4RiCjAt8TUSf4JuEn8mqTK
MfL4Rxf0SEcgEPyJSq1j9AxU3e8ERx5GXj2kJw6lOxIa+Fh0WTcKNImgKz9gMUUL7ls
LnBPghNCCdUvnmYpLhS2HMR1EFrDwIDAQAB"
>
```

and

```
> set type=txt
> _domainkey.listserv.example.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
_domainkey.listserv.example.com       text =

        "o=~"
>
```

To test the functionality of the DKIM DNS entries, you will probably want to use an online service. One such service is MXToolBox.com. They provide a DKIM Lookup tool that is both free and easy to use, located at https://mxtoolbox.com/dkim.aspx . Simply enter the domain name (for example, "listserv.example.com") and the selector (for example, "default") into the two text boxes on that page and click the button for DKIM Lookup. This presents an in-depth report that you can use to verify whether or not your DKIM TXT record has been properly created.

There are many other testers out there; the MXToolBox tester is simply the one that we tend to use at L-Soft. Other testers include https://www.mail-tester.com/spf-dkim-check and https://www.dmarcanalyzer.com/dkim/dkim-check/ ; many more are available via a web search.

Testing DKIM signatures on email

Finally, the simplest way to test that LISTSERV is actually signing emails properly is to create a test list, add an external test account to it (e.g., a GMail account) and send mail to the list.

In the mail headers for the message received by the test account, you'll see something

like the following:

```
DKIM-Signature: v=1; a=rsa-sha256; d=LISTSERV.EXAMPLE.COM; s=DEFAULT;
c=relaxed/relaxed; bh=wCPfXJT/+EjG2NI/0kOFZQI3luKHV0YjC+ZO6gi9sW8=;
i=@LISTSERV.TD.COM;
 h=Date:Sender:From:Subject:To;
 b=b9p4Vj9NSsMxTIRwhO1oRYTYovn8UT/
```

This is the DKIM signature added by LISTSERV as the message was processed and distributed.

Note: The signature will not appear to match either of your DKIM keys; this is because the signature is generated on a per-message basis using the private key in the appropriate *.DKIM file you created and installed earlier.

Above the DKIM-Signature header will be a header showing that the message was properly authenticated:

```
Authentication-Results: mx.google.com;
       dkim=pass header.i=@LISTSERV.EXAMPLE.COM header.s=DEFAULT
header.b=b9p4Vj9N;
       spf=pass (google.com: domain of owner-nolist-test-20160331-
set*john*-doe**gmail*-com@listserv.example.com designates
192.168.6.27 as permitted sender) smtp.mailfrom=owner-nolist-TEST-
20160331-SET*john*-doe**GMAIL*-COM@listserv.example.com;
       dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=example.com
```

Since we used a GMail account, the above is how Google reports the results. In this case, the message not only passed DKIM testing, but also SPF and DMARC (meaning that this particular LISTSERV server is very well provisioned for mail reputation). Other ISPs will produce similar test results, although the formatting may vary depending on the mail product used on the receiving end.

It should be noted that Yahoo, which originally promulgated the now-deprecated DomainKeys standard, will also produce results for their standard, even though LISTSERV uses DKIM:

```
Authentication-Results: mta4010.rog.mail.bf1.yahoo.com
from=listserv.example.com; domainkeys=neutral (no sig);
from= LISTSERV.EXAMPLE.COM; dkim=pass (ok)
```

However, Yahoo is also authenticating against the new DKIM standard, so as long as DKIM gets a “pass”, it doesn’t matter that you did not provide a DomainKeys signature.