

Email Authentication Complete Guide SPF DKIM DMARC ARC BIMI

TechRounder PDF Edition

Live article: <https://www.techrounder.com/data/email-authentication-complete-guide-spf-dkim-dmarc-arc-bimi/>

By Vipin PG | Published March 27, 2026 | Updated March 27, 2026 | Format: Deep Dive | 21 min read

In brief

Email authentication relies on five DNS-based standards that must be deployed in order: SPF authorizes sending servers by IP, DKIM signs messages with cryptographic keys, DMARC aligns SPF and DKIM results with the visible From domain and sets enforcement policy, ARC preserves authentication results through forwarding c.

Last verified: March 2026 • Covers: SPF • DKIM • DMARC • ARC • BIMI

Every email you send passes through a gauntlet of authentication checks before it lands in an inbox - or gets silently dropped. The five standards that run those checks (SPF, DKIM, DMARC, ARC, and BIMI) are documented across a dozen RFCs, scattered vendor help pages, and outdated blog posts that contradict each other. This reference consolidates everything into one place: the full syntax, every parameter, the behavior differences across Gmail / Outlook / Yahoo / Apple Mail, what actually breaks when you misconfigure something, and a deployment checklist you can use today.

No fluff. No "what is an email?" intro. Just the data.

Contents

1. Standards at a Glance - Comparison Table
2. SPF - Sender Policy Framework
3. DKIM - DomainKeys Identified Mail
4. DMARC - Domain-based Message Authentication
5. ARC - Authenticated Received Chain
6. BIMI - Brand Indicators for Message Identification
7. How They Chain Together
8. Major Provider Behavior
9. Common Configuration Mistakes
10. Troubleshooting Quick Reference
11. Deployment Checklist

1. Standards at a Glance

Standard: SPF | RFC: RFC 7208 | What It Verifies: Sending server IP is authorized by the domain owner | DNS Record Type: TXT | Location: '@' (envelope From domain) | Enforcement By: Receiving MTA

Standard: DKIM | RFC: RFC 6376 | What It Verifies: Message was signed by a key the domain owner controls | DNS Record Type: TXT | Location: 'selector._domainkey.domain' | Enforcement By: Receiving MTA

Standard: DMARC | RFC: RFC 7489 | What It Verifies: SPF and/or DKIM aligns with the header From domain; sets policy | DNS Record Type: TXT | Location: '_dmarc.domain' | Enforcement By: Receiving MTA (policy) + domain owner (reports)

Standard: ARC | RFC: RFC 8617 | What It Verifies: Preserves auth results through forwarding and mailing lists | DNS Record Type: TXT (key only) | Location: 'selector._domainkey.domain' | Enforcement By: Forwarding intermediaries

Standard: BIMI | RFC: BIMI Working Group spec | What It Verifies: Displays brand logo in inbox after DMARC passes | DNS Record Type: TXT | Location: 'default._bimi.domain' | Enforcement By: Receiving mail client (display only)

Dependency order matters:

BIMI requires DMARC. DMARC requires SPF and/or DKIM. ARC preserves SPF/DKIM/DMARC results - it does not replace them. You must deploy in order: SPF -> DKIM -> DMARC -> (ARC if needed) -> BIMI.

2. SPF - Sender Policy Framework RFC 7208

SPF lets you publish a list of IP addresses and hostnames that are authorized to send email for your domain. The receiving server checks this list against the IP address of the server that actually delivered the message.

DNS Record Format

v=spf1 [mechanisms] [modifiers] [all]

Publish as a TXT record on your bare domain (e.g. 'example.com'). There must be exactly one SPF record per domain. Multiple SPF TXT records cause a PermError.

Mechanisms - Complete Reference

Mechanism: 'all' | Syntax: 'all' | What It Matches: Matches everything - always placed last | DNS Lookups Used: 0 | Notes: Controls catch-all behavior. Combine with qualifier.

Mechanism: 'ip4' | Syntax: 'ip4:1.2.3.4' or 'ip4:1.2.3.0/24' | What It Matches: Specific IPv4 address or CIDR range | DNS Lookups Used: 0 | Notes: Most efficient mechanism. Use for static sending IPs.

Mechanism: 'ip6' | Syntax: 'ip6:2001:db8::1' or 'ip6:2001:db8::/32' | What It Matches: Specific IPv6 address or CIDR range | DNS Lookups Used: 0 | Notes: Same as ip4 but for IPv6. No DNS lookup cost.

Mechanism: 'a' | Syntax: 'a' or 'a:otherdomain.com' | What It Matches: A/AAAA records of the domain (or specified domain) | DNS Lookups Used: 1 | Notes: Matches if sending IP equals any A record for the domain.

Mechanism: 'mx' | Syntax: 'mx' or 'mx:otherdomain.com' | What It Matches: MX record IPs of the domain | DNS Lookups Used: 1 + 1 per MX host | Notes: Use only if your mail servers also send outbound. Counts multiple lookups.

Mechanism: 'include' | Syntax: 'include:_spf.google.com' | What It Matches: Runs SPF check for another domain; match if that check passes | DNS Lookups Used: 1 + recursive | Notes: Most common way to add third-party senders. Each include costs at least 1 lookup.

Mechanism: 'exists' | Syntax: 'exists:%{i}.dnsbl.example.com' | What It Matches: Passes if an A record exists at the constructed domain | DNS Lookups Used: 1 | Notes: Advanced use only - dynamic allowlisting via DNS.

Mechanism: 'ptr' | Syntax: 'ptr' or 'ptr:domain.com' | What It Matches: Reverse DNS of sending IP matches the domain | DNS Lookups Used: Many | Notes: Deprecated - RFC 7208 §5.5 explicitly discourages use. Avoid entirely.

Qualifiers

Every mechanism can be prefixed with a qualifier that controls the result when the mechanism matches:

Qualifier: Pass | Symbol: '+' (default - can be omitted) | Result: Pass | Practical Effect: Authorized sender. Mail proceeds normally.

Qualifier: Fail | Symbol: '-' | Result: Fail | Practical Effect: Explicitly unauthorized. Most receivers reject or mark spam.

Qualifier: SoftFail | Symbol: '~' | Result: SoftFail | Practical Effect: Probably unauthorized. Accepted but typically marked or lower-scored.

Qualifier: Neutral | Symbol: '?' | Result: Neutral | Practical Effect: No assertion either way. Treated like no SPF record by most receivers.

Modifiers

Modifier: 'redirect' | Syntax: 'redirect=_spf.example.com' | Purpose: Replaces the entire current record with the SPF of another domain | Notes: Cannot be used with 'all'. Useful for managing one master SPF record across domains.

Modifier: 'exp' | Syntax: 'exp=explain.example.com' | Purpose: Provides a human-readable explanation for failures | Notes: Rarely used. The TXT at the specified domain is returned to senders on Fail.

The 10-Lookup Limit - Critical

Hard limit: SPF processing must not require more than 10 DNS lookups.

Mechanisms that consume lookups:

```
a
,
mx
,
include
,
exists
,
redirect
```

. Mechanisms that do NOT consume lookups:

```
ip4
,
ip6
,
all
```

. If the 10-lookup limit is exceeded, the result is

PermError

, which many receivers treat the same as Fail. This is the single most common SPF misconfiguration for companies using 3+ email services (CRM, marketing, transactional, support, etc.).

Each 'include:' costs 1 lookup plus whatever the included record itself requires. A chain of 'include:' references can exhaust the limit quickly without you realizing it. To audit your current count, use MXToolbox SPF Analyzer or 'dig TXT _spf.yourvendor.com' manually and count recursively.

Result Codes - Full Reference

Result: Pass | Meaning: Sending IP is authorized | Typical Receiver Behavior: Accept; positive signal for reputation

Result: Fail | Meaning: Sending IP is explicitly not authorized | Typical Receiver Behavior: Reject or aggressive spam marking

Result: SoftFail | Meaning: Sending IP probably not authorized | Typical Receiver Behavior: Accept but mark; used during transitions

Result: Neutral | Meaning: Domain makes no assertion | Typical Receiver Behavior: Treated same as None

Result: None | Meaning: No SPF record found | Typical Receiver Behavior: No SPF signal; other factors decide

Result: TempError | Meaning: DNS lookup failed temporarily | Typical Receiver Behavior: Most receivers defer (try again later)

Result: PermError | Meaning: Record is invalid or lookup limit exceeded | Typical Receiver Behavior: Varies - often treated as Fail; always investigate

Real-World SPF Examples

Basic - Google Workspace only

```
v=spf1 include:_spf.google.com ~all
```

Multiple providers (Google + Mailchimp + Salesforce)

```
v=spf1 include:_spf.google.com include:servers.mcsv.net include:_spf.salesforce.com ~all
```

Mixed - IPs + includes (most common production scenario)

```
v=spf1 ip4:203.0.113.10 ip4:198.51.100.0/24 include:_spf.google.com include:sendgrid.net -all
```

Subdomain that sends no email

```
v=spf1 -all
```

Publish this on subdomains that never send email (e.g. 'noreply.example.com') to block spoofing attempts on those subdomains.

SPF and Forwarding - The Fundamental Weakness

SPF checks the envelope From (also called MAIL FROM or Return-Path) - not the 'From:' header the user sees. When an email is forwarded, the forwarding server's IP is substituted but the envelope domain stays the same. This means SPF will fail for legitimately forwarded mail. This is exactly why DKIM and ARC exist.

3. DKIM - DomainKeys Identified Mail RFC 6376

DKIM adds a digital signature to outgoing messages. The sending server signs the email using a private key, and publishes the corresponding public key in DNS. Receiving servers verify the signature. Unlike SPF, DKIM survives email forwarding because it signs the message content itself.

How the Signature Works

The sending server computes a hash of selected headers and the message body, encrypts that hash with the private key, and inserts the result as a 'DKIM-Signature:' header. The receiving server retrieves the public key from DNS, decrypts the signature, and compares the hash to what it computes independently from the received message.

DKIM-Signature Header - All Tags Explained

Tag: 'v=' | Name: Version | Required: Yes | Values / Format: '1' | Notes: Always 1. No other version exists in production.

Tag: 'a=' | Name: Algorithm | Required: Yes | Values / Format: 'rsa-sha256' • 'ed25519-sha256' • 'rsa-sha1' | Notes: 'rsa-sha256' is the standard. 'ed25519-sha256' is newer, smaller, and faster - limited receiver support as of 2026. 'rsa-sha1' is deprecated - do not use.

Tag: 'c=' | Name: Canonicalization | Required: No (default: 'simple/simple') | Values / Format: 'relaxed/relaxed' • 'relaxed/simple' • 'simple/simple' • 'simple/relaxed' | Notes: Format: header/body. 'relaxed' tolerates whitespace changes; 'simple' does not. Use 'relaxed/relaxed' in almost all cases - many MTAs rewrite whitespace in transit.

Tag: 'd=' | Name: Signing Domain | Required: Yes | Values / Format: e.g. 'example.com' | Notes: Must match or be an organizational parent of the 'From:' header domain for DMARC alignment.

Tag: 's=' | Name: Selector | Required: Yes | Values / Format: e.g. 'google', 's1', '202401' | Notes: Used to locate the public key in DNS: '{s}._domainkey.{d}'. Use date-based selectors (e.g. '20240101') to make key rotation traceable.

Tag: 'h=' | Name: Signed Headers | Required: Yes | Values / Format: Colon-separated list, e.g. 'from:to:subject:date' | Notes: Must always include 'From'. Also include 'Subject', 'To', 'Date', 'Content-Type', 'MIME-Version'. Do NOT include headers that intermediaries add (e.g. 'Received').

Tag: 'bh=' | Name: Body Hash | Required: Yes | Values / Format: Base64-encoded hash of the message body | Notes: Computed after canonicalization. If the body is modified in transit, this will not match and DKIM fails.

Tag: 'b=' | Name: Signature | Required: Yes | Values / Format: Base64-encoded cryptographic signature | Notes: The actual signature value. Spans multiple lines in headers.

Tag: 't=' | Name: Signature Timestamp | Required: No | Values / Format: Unix timestamp | Notes: When the signature was created. Recommended but not enforced by most receivers.

Tag: 'x=' | Name: Signature Expiry | Required: No | Values / Format: Unix timestamp | Notes: After this time, the signature should be considered invalid. Rarely used in practice.

Tag: 'l=' | Name: Body Length | Required: No | Values / Format: Integer | Notes: Security risk - limits signing to the first N bytes of the body, allowing attackers to append unsigned content. Avoid using this tag.

Tag: 'i=' | Name: Identity | Required: No | Values / Format: e.g. '@example.com' | Notes: User or agent identifier. Rarely used; 'd=' is what matters for DMARC alignment.

Tag: 'z=' | Name: Copied Headers | Required: No | Values / Format: Header field list | Notes: Debug tag - copies original header values into the signature. Only useful for diagnostic tools.

DNS Public Key Record - All Tags Explained

Publish as a TXT record at: '{selector}._domainkey.{domain}'

Tag: 'v=' | Name: Version | Required: Recommended | Values: 'DKIM1' | Notes: If present, must be first tag. Some validators require it.

Tag: 'k=' | Name: Key Type | Required: No (default: 'rsa') | Values: 'rsa' • 'ed25519' | Notes: Must match the 'a=' algorithm in the signature.

Tag: 'p=' | Name: Public Key | Required: Yes | Values: Base64-encoded public key | Notes: If set to empty ('p='), the key is revoked - all signatures using this selector will fail immediately. Use this to revoke compromised keys.

Tag: 'h=' | Name: Acceptable Algorithms | Required: No | Values: Colon-separated list: 'sha256' | Notes: Restricts which hash algorithms may be used with this key.

Tag: 's=' | Name: Service Type | Required: No (default: '*') | Values: 'email' • '*' | Notes: Use 'email' to restrict this key to email only.

Tag: 't=' | Name: Flags | Required: No | Values: 'y' (testing mode) • 's' (strict 'i=' alignment) | Notes: 't=y' tells receivers this domain is testing DKIM; some may skip enforcement. Remove before production.

Tag: 'n=' | Name: Notes | Required: No | Values: Human-readable text | Notes: Purely informational. Ignored by mail systems.

Key Size Recommendations

Key Size: 512-bit RSA | Status: Do Not Use | Notes: Cryptographically broken. Some DNS providers still default to this.

Key Size: 1024-bit RSA | Status: Minimum | Notes: Technically valid but considered weak. Google has deprecated 1024-bit for Gmail signing.

Key Size: 2048-bit RSA | Status: Recommended | Notes: Current best practice. Widely supported by all mail providers and DNS. Use this.

Key Size: 4096-bit RSA | Status: Optional | Notes: Stronger but can exceed DNS TXT record size limits (512 bytes per string). Requires splitting the key across multiple strings in DNS. Use only if your DNS provider handles this correctly.

Key Size: Ed25519 | Status: Emerging | Notes: Compact, fast, and cryptographically strong. Not yet universally supported - some older receivers ignore it entirely. Best practice: publish both an RSA 2048-bit and an Ed25519 selector simultaneously.

Key Rotation Best Practice

DKIM keys should be rotated periodically (at minimum annually; every 6 months for high-volume senders). Rotation process: (1) generate new key pair, (2) publish new public key in DNS under a new selector, (3) wait for DNS propagation, (4) configure your mail server to sign with the new selector, (5) keep the old selector active for at least 72 hours to allow in-transit messages to be verified, (6) revoke old key by setting 'p=' (empty).

What Breaks DKIM in Transit

- Body modifications by mailing list software (adding footers, unsubscribe links)
- Content-Transfer-Encoding changes (base64 <-> quoted-printable)
- Header rewriting by some MTAs (adding/removing trailing whitespace)
- Line ending normalization (CRLF vs LF) - this is why 'relaxed' canonicalization is strongly recommended
- S/MIME or PGP re-signing at a gateway

4. DMARC - Domain-based Message Authentication, Reporting & Conformance RFC 7489

DMARC ties SPF and DKIM together and gives the domain owner control over what happens when they fail. It also generates reports sent back to the domain owner, which is how you learn who is sending as your domain.

Core Concept: Alignment

DMARC does not just check whether SPF or DKIM passed - it checks whether the domain that passed SPF or DKIM aligns with the 'From:' header domain (the domain users actually see). This closes the gap where a spammer could pass SPF on a different domain while spoofing your 'From:' address.

DMARC passes if at least one of the following is true:

- SPF passes AND the envelope From domain aligns with the header From domain
- DKIM passes AND the 'd=' domain in the DKIM signature aligns with the header From domain

DNS Record Format

v=DMARC1; p=none; rua=mailto:reports@example.com;

Publish as a TXT record at: '_dmarc.yourdomain.com'

All DMARC Tags - Complete Reference

Tag: 'v=' | Required: Yes | Default: - | Values: 'DMARC1' | Explanation: Must be first tag, must be exactly 'DMARC1'.

Tag: 'p=' | Required: Yes | Default: - | Values: 'none' • 'quarantine' • 'reject' | Explanation: Policy for messages that fail DMARC. See policy table below.

Tag: 'sp=' | Required: No | Default: Inherits 'p=' | Values: 'none' • 'quarantine' • 'reject' | Explanation: Separate policy for subdomains. If absent, subdomains inherit the organizational domain policy.

Tag: 'adkim=' | Required: No | Default: 'r' | Values: 'r' (relaxed) • 's' (strict) | Explanation: DKIM alignment mode. Relaxed: DKIM 'd=' can be the same domain or a subdomain of the From domain. Strict: exact match required.

Tag: 'aspf=' | Required: No | Default: 'r' | Values: 'r' (relaxed) • 's' (strict) | Explanation: SPF alignment mode. Relaxed: envelope From can be a subdomain. Strict: exact match required.

Tag: 'pct=' | Required: No | Default: '100' | Values: Integer 1-100 | Explanation: Percentage of failing messages the policy applies to. Used to roll out 'quarantine'/'reject' gradually. Note: Google and Yahoo announced plans to ignore 'pct' below 100 for large senders - treat 100 as the only reliable value.

Tag: 'rua=' | Required: No (strongly recommended) | Default: - | Values: Comma-separated mailto URIs | Explanation: Destination for aggregate reports (XML, one report per day per sender). These reports show you all IPs sending as your domain and their pass/fail rates.

Tag: 'ruf=' | Required: No | Default: - | Values: Comma-separated mailto URIs | Explanation: Destination for forensic/failure reports (individual message samples). Many providers no longer send these due to privacy concerns. Gmail does not send 'ruf' reports.

Tag: 'fo=' | Required: No | Default: '0' | Values: '0' • '1' • 'd' • 's' | Explanation: Failure reporting options. '0': report if both SPF and DKIM fail. '1': report if either fails. 'd': report if DKIM fails. 's': report if SPF fails. Colon-separated for multiple: 'fo=1:d:s'.

Tag: 'ri=' | Required: No | Default: '86400' | Values: Integer (seconds) | Explanation: Requested aggregate report interval. Default is 86400 (24 hours). Most senders ignore this and send daily regardless.

Tag: 'rf=' | Required: No | Default: 'afrf' | Values: 'afrf' | Explanation: Failure report format. Only 'afrf' (Auth Failure Report Format, RFC 6591) is defined. No practical reason to specify this tag.

Policy Values Explained

Policy: 'p=none' | Effect on Failing Mail: No action - mail is delivered as normal. Reports still sent if 'rua' is set. | Use Case: Initial monitoring phase. Deploy this first and analyze reports for 2-4 weeks before advancing. | Risk Level: Low - nothing breaks

Policy: 'p=quarantine' | Effect on Failing Mail: Failing mail delivered to spam/junk folder | Use Case: Intermediate enforcement. Move to this after 'none' shows clean reports. | Risk Level: Medium - legitimate mail may be junked

Policy: 'p=reject' | Effect on Failing Mail: Failing mail rejected at the SMTP level - not delivered at all | Use Case: Full enforcement. Move here when you are confident all legitimate senders pass DMARC. | Risk Level: High - misconfigured senders will lose mail

Reporting to a Third-Party Domain

If your 'rua=' address uses a different domain than the policy domain (e.g. your policy is on 'example.com' but reports go to 'reports@dmarc-service.com'), the receiving domain must publish a DNS record authorizing this. Without it, most senders will not deliver the reports.

The receiving domain must publish a TXT record at:
'example.com._report._dmarc.dmarc-service.com'

v=DMARC1;

This tells senders that 'dmarc-service.com' consents to receiving DMARC reports on behalf of 'example.com'.

Subdomain Handling

A DMARC record at '_dmarc.example.com' covers 'example.com' and all subdomains unless overridden. You can set a stricter subdomain policy with 'sp=reject' without changing the top-level policy, or publish a separate '_dmarc.sub.example.com' record to override for a specific subdomain.

Recommended Rollout Sequence

```
# Step 1 - Monitor (publish immediately, no risk)
v=DMARC1; p=none; rua=mailto:dmarc@yourdomain.com;

# Step 2 - After 2-4 weeks of clean reports: quarantine a small percentage
v=DMARC1; p=quarantine; pct=10; rua=mailto:dmarc@yourdomain.com;

# Step 3 - Increase gradually
v=DMARC1; p=quarantine; pct=50; rua=mailto:dmarc@yourdomain.com;

# Step 4 - Full quarantine
v=DMARC1; p=quarantine; pct=100; rua=mailto:dmarc@yourdomain.com;

# Step 5 - Full enforcement (goal state)
v=DMARC1; p=reject; rua=mailto:dmarc@yourdomain.com; sp=reject;
```

5. ARC - Authenticated Received Chain RFC 8617

ARC solves a specific problem: SPF always fails for forwarded mail, and DKIM often fails when mailing lists modify the message body. DMARC then fails as a consequence, and legitimate mail gets rejected. ARC provides a way for intermediaries to say "I received this message, it passed authentication when I received it, and here is my signed attestation of that fact."

When ARC Is Relevant

- Email forwarding services (e.g. university alumni addresses, custom domain forwarding)
- Mailing lists that add footers or subject prefixes
- Security gateways that modify headers
- Compliance/archiving systems that relay mail

ARC is added by intermediaries (forwarders, mailing list servers), not by the original sender. If you operate a forwarding service or mailing list, you should implement ARC. If you only send and receive normal direct email, you do not need to configure ARC - receivers handle it automatically when processing forwarded messages.

The Three ARC Headers

Each intermediary that handles the message adds a set of three headers, identified by an instance number ('i=') that increments with each hop:

Header: 'ARC-Authentication-Results' | Abbreviation: AAR | Contains: Copy of the 'Authentication-Results' header as seen at this hop, with instance number | Purpose: Records what the authentication state was when this intermediary received the message

Header: 'ARC-Message-Signature' | Abbreviation: AMS | Contains: A DKIM-like signature of the message (headers + body) at this hop | Purpose: Proves the message content was unchanged between this hop and the next receiver

Header: 'ARC-Seal' | Abbreviation: AS | Contains: Signature covering all prior ARC headers plus this hop's AAR and AMS | Purpose: Creates a tamper-evident chain - verifying the seal confirms the entire ARC chain is intact

ARC Key Tags

The 'ARC-Seal' and 'ARC-Message-Signature' headers use the same tag structure as DKIM signatures, with these additional/modified tags:

Tag: 'i=' | Meaning: Instance number | Notes: Integer 1-50. Increments by 1 for each intermediary. If a chain exceeds 50 hops, ARC processing stops.

Tag: 'cv=' | Meaning: Chain validation result (ARC-Seal only) | Notes: 'none' (first hop, no prior chain) • 'pass' (prior chain valid) • 'fail' (prior chain invalid - chain is broken and cannot be trusted)

ARC is advisory, not mandatory.

Receiving servers can choose to honor ARC results or ignore them. Gmail and Microsoft 365 both use ARC results as a positive signal when deciding whether to deliver forwarded mail. Receivers are never required to accept mail just because ARC passes - it is one factor among many in the spam/reputation evaluation.

ARC DNS Record

ARC uses the same DNS TXT record format as DKIM for its public keys, published at the same location: '{selector}._domainkey.{intermediary-domain}'. If you operate a forwarding service, you need to generate an ARC key pair and publish the public key the same way you would for DKIM.

6. BIMI - Brand Indicators for Message Identification

BIMI allows organizations with properly authenticated email to display their brand logo directly in the inbox beside the sender's name - before the message is opened. Supported clients show a verified avatar in place of the generic initial circle.

Requirements (All Must Be Met)

1. DMARC policy of 'p=quarantine' or 'p=reject' (not 'none') with 'pct=100'
2. SVG logo file in SVG Tiny PS (Portable/Secure) format - not standard SVG
3. Logo hosted at a publicly accessible HTTPS URL
4. A Verified Mark Certificate (VMC) - required by Gmail and Yahoo for the verified checkmark indicator (not required to show a logo in all providers)
5. Trademark registration for the logo in at least one jurisdiction (required to obtain a VMC)

DNS Record Format

v=BIMI1; l=https://brand.example.com/logo.svg; a=https://brand.example.com/vmc.pem;

Publish as a TXT record at: 'default._bimi.yourdomain.com'

All BIMI DNS Record Tags

Tag: 'v=' | Required: Yes | Value: 'BIMI1' | Notes: Must be first.

Tag: 'l=' | Required: Yes (set to empty if no logo) | Value: HTTPS URL to SVG Tiny PS file | Notes: If set to empty ('l='), you are explicitly opting out of BIMi for this domain. Max URL length: 2048 chars.

Tag: 'a=' | Required: No (required for verified checkmark) | Value: HTTPS URL to VMC '.pem' file | Notes: VMC is a certificate issued by a Certificate Authority (currently: DigiCert, Entrust). Required for the blue checkmark in Gmail and Yahoo.

SVG Logo Requirements (SVG Tiny PS)

Standard SVG files will not work. The SVG must conform to SVG Tiny Portable/Secure, which is a restricted subset of SVG 1.2:

- File must begin with the correct XML declaration and SVG namespace
- No JavaScript, no external resources, no raster images embedded
- No '<use>' elements that reference external files
- Must have a square 'viewBox' (equal width and height)
- No '<script>' or '<foreignObject>' elements
- Colors must use SVG named colors or hex values - no CSS variables
- File size: typically under 32KB; no hard limit defined but large files slow verification
- The logo must be centered and fill the square frame with a solid background (no transparency)

Verified Mark Certificate (VMC)

A VMC is an extended validation certificate that cryptographically binds your logo to your domain and trademark. It is issued by approved Certificate Authorities after identity and trademark verification.

Aspect: Current approved CAs | Detail: DigiCert, Entrust

Aspect: Trademark requirement | Detail: Logo must be registered with a recognized trademark office (USPTO, EUIPO, JPO, UK IPO, CIPO, IP Australia, and others)

Aspect: Certificate validity | Detail: Typically 1 year; must be renewed

Aspect: Cost | Detail: Approximately \$1,000-\$1,500/year per domain depending on CA and issuer

Aspect: File format | Detail: PEM-encoded certificate hosted at the URL specified in 'a='

Aspect: Without VMC | Detail: Some providers (Apple Mail, Fastmail) show the logo without a VMC. Gmail and Yahoo require VMC for the verified indicator.

BIMI for Subdomains

BIMI applies per domain/subdomain. A record at 'default._bimi.example.com' does not automatically apply to 'mail.example.com'. Each sending domain needs its own BIMI record, DMARC policy, and logo URL.

7. How They Chain Together - End-to-End Flow

Here is the exact sequence a receiving mail server runs when it accepts an inbound message:

SMTP connection received from IP: 203.0.113.45

Envelope From (Return-Path): bounce@mail.example.com

Header From: alice@example.com

Subject: Your invoice

Step 1 - SPF check

Query: TXT record for mail.example.com

Result: 203.0.113.45 matches ip4:203.0.113.0/24

SPF result: PASS

Alignment: mail.example.com vs example.com -> relaxed match -> ALIGNED

Step 2 - DKIM check

Read: DKIM-Signature header (d=example.com, s=20240101)

Query: TXT record for 20240101._domainkey.example.com

Verify signature against message content

DKIM result: PASS

Alignment: d=example.com vs From: example.com -> exact match -> ALIGNED

Step 3 - DMARC check

Query: TXT record for _dmarc.example.com

Policy: p=reject

SPF passed AND aligned? YES -> DMARC identifier = SPF

DKIM passed AND aligned? YES -> additional positive signal

DMARC result: PASS

Action: deliver normally

Step 4 - ARC check (only if message was forwarded)

Not applicable - direct delivery

Step 5 - BIMI check (if DMARC passed and client supports BIMI)

Query: TXT record for default._bimi.example.com

l= URL: fetch SVG logo

a= URL: fetch and validate VMC

Display logo in inbox UI

DMARC Failure Scenarios

Scenario: Direct send, all configured correctly | SPF: Pass | DKIM: Pass | DMARC Aligned?: Yes | Result: DMARC Pass

Scenario: Third-party sender not in SPF, no DKIM | SPF: Fail | DKIM: None | DMARC Aligned?: No | Result: DMARC Fail

Scenario: Forwarded mail (SPF fails, DKIM intact) | SPF: Fail | DKIM: Pass | DMARC Aligned?: DKIM aligns | Result: DMARC Pass (DKIM saves it)

Scenario: Mailing list (SPF fails, DKIM broken by footer) | SPF: Fail | DKIM: Fail | DMARC Aligned?: No | Result: DMARC Fail (ARC needed)

Scenario: SPF passes but wrong domain (subdomain mismatch, strict mode) | SPF: Pass | DKIM: Pass | DMARC Aligned?: No (strict mismatch) | Result: DMARC Fail

Scenario: DKIM passes but 'd=' is different root domain | SPF: Pass | DKIM: Pass | DMARC Aligned?: No (different org) | Result: DMARC Fail (DKIM not aligned)

8. Major Provider Behavior

? Gmail / Google Workspace

- Fully enforces DMARC 'p=reject'
- Sends aggregate ('rua') reports daily; does NOT send forensic ('ruf') reports
- Honors ARC for forwarded mail from trusted intermediaries
- Requires VMC for verified BIMI checkmark (blue checkmark in avatar)
- Shows BIMI logo in Gmail web and Android/iOS apps
- Uses its own 'Authentication-Results' header with 'compauth' composite score for UI signals
- Publishes own SPF and DKIM for outbound mail via Google servers

? Microsoft 365 / Outlook.com

- Enforces DMARC; adds its own 'compauth' (composite authentication) layer
- Sends aggregate reports; no forensic reports

- Has anti-spoofing intelligence that can override DMARC - mail from trusted IPs may pass even with weak auth
- BIMI support announced - limited rollout as of early 2026; VMC required for checkmark
- Honors ARC headers from known ARC signers
- Implicit authentication: if a sender has good reputation history with Microsoft, failures may be tolerated
- EOP (Exchange Online Protection) applies additional ML-based spam scoring on top of auth results

? Yahoo / Yahoo Mail

- One of the earliest adopters of DMARC enforcement
- Fully enforces 'p=reject'
- Sends aggregate reports; forensic report delivery inconsistent
- BIMI supporter - requires VMC for blue verified indicator
- Yahoo announced in 2024 that bulk senders must have DMARC at minimum 'p=none' with 'rua=' set
- ARC support present in their infrastructure

? Apple Mail / iCloud Mail

- Honors SPF, DKIM, and DMARC for filtering decisions
- BIMI logo display supported in Apple Mail (iOS 16+, macOS Ventura+) without VMC requirement - logo shown based on DMARC pass alone
- Does NOT send DMARC aggregate reports from iCloud Mail
- Mail Privacy Protection (iOS 15+) pre-fetches content via Apple proxy - does not affect authentication but breaks open-tracking
- ARC support present

Bulk Sender Requirements (2024 Mandates)

In February 2024, Google and Yahoo jointly announced requirements for senders sending more than 5,000 messages per day to Gmail/Yahoo addresses:

Requirement: SPF or DKIM (at minimum) | Google: Required | Yahoo: Required

Requirement: DMARC record present | Google: Required ('p=none' minimum) | Yahoo: Required ('p=none' minimum)

Requirement: Valid PTR record for sending IPs | Google: Required | Yahoo: Required

Requirement: TLS for SMTP connections | Google: Required | Yahoo: Required

Requirement: One-click List-Unsubscribe (RFC 8058) | Google: Required for marketing mail | Yahoo: Required for marketing mail

Requirement: Spam rate threshold | Google: Under 0.10% (warned at 0.08%) | Yahoo: Enforced but threshold not published

9. Common Configuration Mistakes

Mistake: Multiple SPF TXT records on same domain | Standard: SPF | Effect: PermError - both records ignored, all mail fails SPF | Fix: Merge all mechanisms into one TXT record

Mistake: Exceeding 10 DNS lookup limit | Standard: SPF | Effect: PermError - treated as Fail by many receivers | Fix: Replace 'include:' chains with 'ip4:'/'ip6:' where possible; use SPF flattening services

Mistake: Using '+all' (pass all) | Standard: SPF | Effect: Any server on the internet can send as your domain and pass SPF - no protection whatsoever | Fix: Use '~all' or '-all'

Mistake: Not adding new sending services to SPF | Standard: SPF | Effect: Legitimate mail from new service fails SPF, DMARC, and may be rejected | Fix: Update SPF record before activating any new email-sending service

Mistake: DKIM using 'simple' canonicalization | Standard: DKIM | Effect: Signatures break on any whitespace modification in transit | Fix: Use 'c=relaxed/relaxed'

Mistake: DKIM not signing the 'From' header | Standard: DKIM | Effect: DKIM cannot align with DMARC - DMARC alignment fails | Fix: Always include 'from' in 'h=' tag

Mistake: DKIM 1024-bit key | Standard: DKIM | Effect: Considered weak; some providers downgrade reputation signal | Fix: Rotate to 2048-bit RSA key

Mistake: Never rotating DKIM keys | Standard: DKIM | Effect: Compromised key can be used indefinitely to forge signed mail | Fix: Rotate at least annually; use dated selectors

Mistake: Going directly to 'p=reject' without monitoring | Standard: DMARC | Effect: Legitimate sending sources not in SPF/DKIM will lose all mail silently | Fix: Start with 'p=none' + 'rua=', monitor for 2-4 weeks, then advance

Mistake: No 'rua=' in DMARC record | Standard: DMARC | Effect: You receive zero reports - cannot discover unauthorized senders or misconfigurations | Fix: Always set 'rua=' even at 'p=none'

Mistake: DMARC 'sp=' not set - subdomains left open | Standard: DMARC | Effect: Attackers can spoof 'anything.yourdomain.com' while your main domain is protected | Fix: Add 'sp=reject' once main domain is clean

Mistake: BIMI with DMARC 'p=none' | Standard: BIMI | Effect: BIMI will not display - 'none' policy is insufficient | Fix: Move DMARC to 'quarantine' or 'reject' before deploying BIMI

Mistake: BIMI SVG not in SVG Tiny PS format | Standard: BIMI | Effect: Logo rejected by Gmail, Yahoo - not displayed | Fix: Regenerate logo in SVG Tiny PS; use W3C validator to confirm

Mistake: Setting 'l=' tag in DKIM | Standard: DKIM | Effect: Attackers can append unsigned content to messages that appear DKIM-signed | Fix: Never use the 'l=' body length tag

Mistake: Publishing DMARC only on primary domain, not on parked/defensive domains | Standard: DMARC | Effect: Parked domains ('v=spf1 -all' only) can still be used for spoofing if no DMARC | Fix: Publish 'v=DMARC1; p=reject;' on all domains you own that never send email

10. Troubleshooting Quick Reference

Symptom: Mail going to spam despite SPF passing | Most Likely Cause: DMARC not configured, or SPF not aligned with header From domain | How to Diagnose: Check 'Authentication-Results' header in received message; look for 'dmarc=fail'

Symptom: SPF PermError in headers | Most Likely Cause: Multiple SPF records, or DNS lookup limit exceeded | How to Diagnose: Run 'dig TXT yourdomain.com' - count TXT records starting with 'v=spf1'; use MXToolbox SPF lookup to count lookups

Symptom: DKIM signature failing for some recipients but not others | Most Likely Cause: Body modification in transit (mailing list footer, encoding change) | How to Diagnose: Check if failing recipients are mailing lists or forwarders; confirm 'c=relaxed/relaxed' is set

Symptom: DKIM failing for all recipients after a recent change | Most Likely Cause: Private key mismatch with published public key, or wrong selector | How to Diagnose: Query 'selector._domainkey.yourdomain.com' - confirm the public key matches your private key

Symptom: DMARC aggregate reports show unknown sending IPs | Most Likely Cause: A service is sending as your domain that is not in your SPF record | How to Diagnose: Identify the IP using ARIN/RIPE WHOIS; add to SPF or investigate if unauthorized

Symptom: DMARC reports stopped arriving | Most Likely Cause: 'rua=' mailbox full, or third-party reporting domain authorization record missing | How to Diagnose: Check mailbox; if using external reporting service, verify 'yourdomain.com._report._dmarc.reportingdomain.com' TXT record exists

Symptom: Forwarded mail failing DMARC at destination | Most Likely Cause: SPF fails (new IP), DKIM broken by forwarder modification, ARC not implemented by forwarder | How to Diagnose: Check if forwarder implements ARC; if not, contact forwarder or use SRS (Sender Rewriting Scheme) for SPF

Symptom: BIMI logo not displaying in Gmail | Most Likely Cause: DMARC not at 'p=quarantine/reject', VMC missing or invalid, SVG not Tiny PS format | How to Diagnose: Use Google Postmaster Tools; verify DMARC policy; validate SVG at W3C; check VMC expiry

Symptom: Mail rejected with "550 5.7.1 SPF check failed" | Most Likely Cause: Sending IP not in SPF record for the envelope From domain | How to Diagnose: Check which IP is sending and which domain is in Return-Path/MAIL FROM; add IP to SPF or fix sending configuration

Symptom: Mail rejected with "550 5.7.26 This message does not have authentication information" | Most Likely Cause: Google-specific: message lacks any DMARC-aligned SPF or DKIM (2024 bulk sender requirement) | How to Diagnose: Ensure both SPF and DKIM are configured and aligned; confirm DMARC record exists

11. Deployment Checklist

SPF

- Exactly one TXT record starting with 'v=spf1' on each sending domain
- All outbound mail servers and services included (own IPs + all third-party services)
- DNS lookup count verified to be 10 or fewer
- Record ends with '~all' or '-all' (never '+all')
- Subdomains that do not send mail have 'v=spf1 -all' published
- Parked/defensive domains have 'v=spf1 -all' published

DKIM

- At least one DKIM key pair generated and published (2048-bit RSA minimum)
- Signing configured on all outbound mail streams
- Canonicalization set to 'relaxed/relaxed'
- 'From' header included in signed headers ('h=' tag)
- 'l=' body length tag NOT used
- Key rotation schedule documented
- Test email verified - check 'DKIM=pass' in 'Authentication-Results' header

DMARC

- TXT record published at '_dmarc.yourdomain.com'
- 'rua=' set to a monitored mailbox or reporting service
- Started at 'p=none' and monitored for minimum 2 weeks before advancing
- All sending sources identified via aggregate reports
- Advanced to 'p=quarantine' after sources confirmed clean
- Advanced to 'p=reject' as final state
- 'sp=reject' added to prevent subdomain spoofing
- All owned non-sending domains have 'p=reject' DMARC records

ARC (if operating a forwarder or mailing list)

- ARC key pair generated and published in DNS
- Mail server software supports ARC signing (Postfix + OpenARC, rspamd, etc.)
- ARC signing configured and verified in outbound headers

BIMI (optional - after DMARC is at reject/quarantine)

- DMARC at 'p=quarantine' or 'p=reject' with 'pct=100'
- Logo created in SVG Tiny PS format, passes W3C validation
- Logo hosted at a stable HTTPS URL
- VMC obtained (for Gmail/Yahoo verified indicator)

- TXT record published at 'default._bimi.yourdomain.com'

- Display tested in Gmail, Yahoo, Apple Mail

All information in this reference is sourced from the original RFCs (7208, 6376, 7489, 8617), BIMI Working Group specifications, and official provider documentation. Standards in this area evolve - check the relevant RFC or provider documentation for edge cases and recent amendments.