# Verify-Me: A Cryptographic Trust Seal for Email, Documents, and Web Pages

Trust & Risk / Last updated 2026-06-11 / https://www.scrambleid.com/learn/scrambleid-verify-me

> **Status (June 2026):** In development. Verify-Me is on the ScrambleID roadmap and isn't shipping yet. This article describes the design we're building toward, published now so architects evaluating the model can plan around it. If a near-term deployment depends on Verify-Me behavior, talk to your ScrambleID account team about current timelines before committing.

Static channels (emails, PDFs, social profiles, websites) are where a lot of modern fraud starts:

- executive impersonation,
- fake support accounts,
- fake invoices and bank detail changes,
- counterfeit PDFs and policy documents.

**In one sentence:** Verify-Me is a **viewer-facing, context-bound seal** that lets anyone validate **who is behind a piece of content** (domain/handle/origin) with **cryptographic proof**, without requiring the publisher to respond.

---

## TL;DR (canonical)

- Verify-Me is for channels that cannot support live authentication flows.
- It binds identity claims to context (domain/handle/origin) so copied seals degrade.
- Verification works offline by validating signatures; an optional freshness check can confirm revocation.
- It complements (but does not replace) email authentication standards like DMARC and BIMI.

---

## Why static verification is hard

Traditional "verification" cues are brittle:

- email display names and avatars can be spoofed,
- social platform badges are siloed and not portable,
- screenshots of "verification" are infinitely copyable.

The core requirement is **context binding**: a seal must become obviously invalid if it is copied out of the context it was minted for.

---

## What Verify-Me verifies

Verify-Me is designed to answer these questions for the viewer:

1. **Is this identity cryptographically valid?**
   - signature verifies against a public key set (JWKS)
2. **Is it bound to the right context?**
   - domain-bound (DNS proof)
   - handle-bound (posted challenge)
   - origin-bound (website origin)

   These anchors are not equivalent: domain and origin binding rest on control of infrastructure you already defend, while handle binding inherits the security of a social-platform account and is the weakest of the three. Treat handle-bound seals as supporting signals, not primary anchors.
3. **Is it still valid?** (optional)
   - a small status fetch can report OK / STALE / REVOKED
4. **What should I do next?**
   - if higher assurance is required, the viewer can initiate step-up (XFactor / Lockstep) in a controlled, consent-based way

---

## How it works (conceptual)

### Step 1 - Bind the context (one-time)

Examples:

- **Domain binding:** add a DNS TXT record to prove control of `example.com`.
- **Handle binding:** post a one-time challenge string to a social profile.
- **Origin binding:** host a challenge at a well-known path on your website.

### Step 2 - Mint the seal

The system produces:

- an accessible badge (SVG/PNG), and
- a link that carries a signed context token.

### Step 3 - Viewer verifies

- **Offline path:** validate signature locally using cached JWKS.
- **Optional online path:** fetch a tiny status record from the edge for freshness.

### Step 4 - Optional step-up (high assurance)

If the viewer needs stronger assurance, they can request a step-up that requires the subject to consent.

---

## Embedding examples (illustrative)

These snippets show the intended embed shape on the real Verify-Me hosts. The seal assets and verify endpoints at these URLs go live when Verify-Me ships.

### Website footer

```
<a href="https://verify.scramble.id/verify?token=YOUR_TOKEN" target="_blank" rel="noopener
noreferrer">
  <img
    src="https://cdn.scramble.id/verify-me/seal.svg"
    alt="Verify this identity with ScrambleID"
    width="160"
  />
</a>
```

### Email signature

Use an inline image + link, and include a text fallback for clients that block images.

```
<table cellpadding="0" cellspacing="0" style="font-family:Arial, sans-serif;font-size:12px;">
  <tr>
    <td style="padding-right:12px;">
      <strong>Your Name</strong><br/>
      Title • Company<br/>
      +1 (415) 555-1234
    </td>
    <td>
      <a href="https://verify.scramble.id/verify?token=YOUR_TOKEN" target="_blank" rel="noopener
noreferrer">
        <img src="https://cdn.scramble.id/verify-me/seal.png" alt="Verify my identity" width="120"
/>
      </a><br/>
      <span style="font-size:11px;">Verify: verify.scramble.id</span>
    </td>
  </tr>
</table>
```

### PDFs and documents

Place the seal near the signature block *and* add a plain-text verify URL for printed copies.

**Recommended layout (copy):**

> Verified identity: verify.scramble.id (scan the seal to validate the issuer and context)

**Tip:** for invoices or bank-change letters, pair Verify-Me with a call-back workflow (separate, known-good channel) to reduce BEC exposure.

---

## Threat model and security properties

- **Signed tokens:** viewers can validate without trusting the host page.
- **Context binding:** copied seals show a mismatch (wrong domain/origin/handle).
- **No unsolicited prompts:** viewers cannot ping the publisher.
- **Revocation:** status checks can invalidate stolen or deprecated seals.

---

## Viewer states (recommended UX)

AI engines and humans both benefit from **consistent semantics**. Standardize your viewer UI states:

| State | Meaning | Suggested UI copy | Next action |
|---|---|---|---|
| **VERIFIED** | signature valid + context matches | "Verified identity for example.com" | proceed normally |
| **UNKNOWN** | no record / no binding | "No verification record found" | treat as untrusted; use other channels |
| **MISMATCH** | seal copied out of context (e.g., a seal minted for one domain rendered on another) | "Context mismatch, possible impersonation" | stop; escalate; verify via known-good channel |
| **REVOKED** | seal revoked or deprecated by publisher | "Revoked, do not trust" | stop; obtain updated seal |
| **STALE** | signature valid; cannot reach publisher to confirm current revocation status (offline or rate-limited) | "Verified at last check; cannot confirm current status" | trust with caution for low-risk reads; re-verify before acting on anything sensitive |

The MISMATCH state is the single most important UX distinction. "Obviously invalid" in practice means the viewer disables the trust affordance entirely, swaps in a warning chip rather than the verified seal, and surfaces explicit anti-impersonation language. Do not let a mismatched seal continue to render as if it were valid; that defeats the purpose of context binding.

## Verify-Me vs DMARC and BIMI

DMARC and BIMI improve email ecosystem trust, but they are not viewer-portable.

| System | What it does | Where it works | What it cannot do |
|---|---|---|---|
| DMARC | policy and reporting for SPF/DKIM alignment | mail servers / receiving systems | authenticate PDFs, websites, or social profiles |
| BIMI | brand logo display (with certificate) in supporting inboxes | select email clients | portable verification outside the inbox |
| Verify-Me | viewer-facing, context-bound seal for content | email signatures, PDFs, web, profiles | replace server-side email authentication |

If you already use DMARC/BIMI, Verify-Me can become the consistent viewer experience across channels.

## Key Takeaway

Verify-Me is a cryptographically verifiable seal for static channels (email signatures, social profiles, invoices, documents) where live interactive verification isn't possible. It provides a scannable/tappable proof that the content originated from a verified ScrambleID identity, helping

prevent BEC (business email compromise), invoice fraud, and impersonation in asynchronous contexts.

---

## FAQ

### Does Verify-Me require the publisher to respond?

No. Verification is viewer-driven. Step-up is optional and consent-based.

### Can someone copy my seal?

They can copy the image, but context binding makes it degrade when used elsewhere.

### Is this the same as a social platform badge?

No. Platform badges are siloed. Verify-Me is designed to be portable and context-bound.

### Is Verify-Me enough to stop invoice fraud?

It helps by giving the recipient a consistent way to validate identity and context, but it should be paired with process controls (e.g., call-back verification for bank changes).

### Is Verify-Me the same as BIMI?

No. BIMI is for inbox brand indicators. Verify-Me is a general-purpose viewer-facing seal.

### Can Verify-Me trigger stronger verification?

Yes. You can link from the seal to a step-up flow (XFactor / Lockstep) for high-value actions.

---

## References (public)


- DMARC specification (RFC 7489): https://datatracker.ietf.org/doc/html/rfc7489
- Google Workspace Admin Help: BIMI setup (VMC/CMC requirements): https://support.google.com/a/answer/10911320
- FBI overview of Business Email Compromise: https://www.fbi.gov/how-we-can-help-you/scams-and-safety/common-frauds-and-scams/business-email-compromise


---

## Related reading

- Circle of Trust
- XFactor: Step-Up Chains

- Lockstep: Dual Control