Dynamic Identifiers Explained: The Cryptographic Primitive Behind Phishing-Resistant Authentication

Fundamentals / Last updated 2026-06-11 / https://www.scrambleid.com/learn/dynamic-identifiers-did-qid

In one sentence: A Dynamic Identifier (DID) is a **short-lived, single-use challenge** that binds a confirmation to a specific session and intent; a QID is the **signed QR envelope** that carries a DID plus validation metadata.

TL;DR (canonical)

- DIDs are not shared secrets and are not reusable.
- QIDs are signed so scanners can validate authenticity even during key rotation.
- The DID is bound to the correct session (websocket/call session/link session), so attackers cannot reuse a screenshot or forward a proof.
- DID/QID is the reusable primitive across Online (web), Caller (voice), Desktop, and People.

What are DID and QID?

DID (Dynamic Identifier)

A **server-issued** identifier with these properties:

- short TTL (seconds)
- single-use (consumed on success)
- scoped to a session + policy
- safe to read aloud or display (not a secret)

QID (QR Identifier)

A QR-encoded structure that contains:

- a DID
- signature/cert thumbprint metadata
- optional context needed for verification

The purpose is to let the app verify the QR has not been altered.

How is a DID different from an OTP?

Property	DID	OTP (TOTP/SMS/email)
Is it a secret?	No (identifier)	Yes (secret code)
Intended to be forwarded?	No	Often effectively yes
Replay risk	Low (single-use + binding)	Medium-high (phishable/replayable)
Binds to a specific session?	Yes	Usually no
Human intent signal	Often paired with typed code and context	Often just "enter the code"

How do DID and QID work in practice?

> **Cryptographic detail:** the structure of the signed confirmation payload, the verifier's atomic check, recommended TTLs per channel, and the threat model coverage are specified in Session binding cryptography in the architecture reference. This article covers what a DID is; that section covers how the binding works end-to-end.

```
sequenceDiagram
  participant U as User
  participant B as Browser/IVR/Verifier
  participant S as ScrambleID Server
  participant A as Scramble App

  B-»S: Start session
  S--»B: DID (+ QID for QR)
  B--»U: Show/read DID
  U-»A: Scan QID or type DID
  A-»S: Confirm DID + device proof (ZID)
  S--»B: Success bound to session
```

What makes DID/QID phishing-resistant?

A DID system is only as safe as its bindings. ScrambleID achieves phishing resistance by using DID/QID alongside:

- **session binding** (the DID approval must map to the correct websocket/call session)
- **device binding** (approval is tied to an enrolled device ZID)
- **expiry** (very short TTL)
- **single use** (consumed and invalid after success)

This is how ScrambleID prevents:

- screenshot replay (use the DID from a photo)
- link forwarding (approve someone else's session)
- "approve on a fake prompt" patterns

What are common DID failure states?

- **Expired DID:** show a single refresh action; do not fall back to OTP.
- **Wrong DID entered:** show clear retry counter and lockouts.
- **Late confirmation:** treat as failure; do not apply to a new session.
- **Session mismatch:** cancel and log as a potential attack signal.

What should your DID/QID implementation checklist include?

- Use short TTLs (seconds) for DIDs.
- Bind confirmation to the session id (websocket/call) and origin context.
- Use typed-code confirmation where possible (explicit user intent).
- Emit standardized events (presented, confirmed, timeout, mismatch).

Key Takeaway

A Dynamic Identifier (DID) is a short-lived, single-use challenge that binds confirmation to a specific session and intent. Unlike OTPs, DIDs are not secrets, they are identifiers that become safe through binding, expiry, and device proofs. A QID is the signed QR envelope carrying a DID plus validation metadata. This is the reusable primitive across ScrambleID's web, voice, desktop, and people channels.

FAQ

What is a dynamic identifier?

A dynamic identifier (DID in ScrambleID terminology) is a short-lived, single-use code that identifies an authentication session without revealing sensitive information. Unlike passwords or OTPs, a

dynamic identifier is not a secret, it's a session pointer that requires cryptographic confirmation from a bound device to complete authentication.

How is a DID different from a one-time password (OTP)?

An OTP (one-time password) is a secret that grants access when entered correctly. A DID is a session identifier that requires additional proof (device-bound cryptographic signature) to complete. If someone overhears a DID, they cannot use it without the registered device. If someone overhears an OTP, they can use it immediately.

What is a QID?

A QID (QR Identifier) is a signed QR code that encodes a DID plus cryptographic metadata (signature, expiry, session binding). Scanning a QID validates its signature before presenting the confirmation UX. This prevents QR phishing attacks where malicious QRs redirect to fake sites.

Why are dynamic identifiers phishing-resistant?

Dynamic identifiers are phishing-resistant because completion requires: (1) a registered device with bound keys, (2) user confirmation in the trusted app, and (3) session binding that prevents relay attacks. An attacker cannot complete authentication by intercepting the identifier alone.

What is the typical TTL for a DID?

DIDs typically expire in 30-90 seconds depending on the channel, with 90 seconds as the recommended maximum; see the canonical per-channel TTL table in the architecture reference. Shorter is better for security; the upper end of the range exists for accessibility. The TTL should be long enough for the user to complete confirmation but short enough to limit attack windows.

Can DIDs be used across channels?

Yes. The same DID/QID architecture works for web login, phone/IVR verification, in-person verification, and cross-device authentication. This is the foundation of omnichannel authentication.

References (public)

- WebAuthn (W3C): https://www.w3.org/TR/webauthn/
- CISA phishing-resistant MFA fact sheet: https://www.cisa.gov/sites/default/files/publications/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf

---

Related reading

- Phishing-Resistant Web Authentication
- Caller Authentication
- ScrambleID Architecture