@justin _ _ richer

https://bspk.io/

HTTP Signing

What are we solving?

- Message-level signature to HTTP requests
- Signature over headers, body, request elements
- Detached

Why are we solving it?

- Key possession proof
- Authentication
- Request message integrity
- Non-repudiation
- Audit

Why is it hard?

- Detached signatures required for HTTP transparency
  - Encapsulation breaks layers
- HTTP messages get transformed
- Common to parse and re-serialize

What about TLS?

- Proxies and reverse proxies
  - TLS protection stops at the terminator
- Mutual TLS doesn't work for many use cases
  - Certificate deployment issues
- Relayed messages don't work

What about S/HTTP?

- Wraps the entire request in a signed container
- Duplicates or hides HTTP functions
- Nobody's heard of it

What about JOSE?

- Wraps whole request in JOSE object
- Duplicates or hides HTTP functions
- It's SOAP with curly braces
  - Pls no

Options

- Cavage
  - https://tools.ietf.org/html/draft-cavage-http-signatures
- OAuth DPoP
  - https://tools.ietf.org/html/draft-fett-oauth-dpop
- AWSv4
  - https://docs.aws.amazon.com/general/latest/gr/signature-version-4.html
- OAuth PoP
  - https://tools.ietf.org/html/draft-ietf-oauth-signed-http-request
- XYZ
  - https://tools.ietf.org/html/draft-richer-transactional-authz

Options

- Cavage
  - https://tools.ietf.org/html/draft-cavage-http-signatures
- OAuth DPoP
  - https://tools.ietf.org/html/draft-fett-oauth-dpop
- AWSv4
  - https://docs.aws.amazon.com/general/latest/gr/signature-version-4.html
- OAuth PoP
  - https://tools.ietf.org/html/draft-ietf-oauth-signed-http-request
- XYZ
  - https://tools.ietf.org/html/draft-richer-transactional-authz

"Standards"

	Cavage	OAuth DPoP	OAuth PoP	XYZ	AWSv4
Method	Y	Y	Y	N	Y
URL	Y (request)	Y (request)	Y (parsed)	N	Y
Headers	Y	N	Y	N	Y
Body	Y (Digest header)	N	Y	Y	Y
Token	N	N	Y	Y?	Y
Keys	Sym/Asym	Asym	Sym/Asym	Asym	Sym
Draft	ID (External Development)	ID (Pending WG)	WG Draft (Expired)	ID	Proprietary

Cavage Signature Drama

2013
- draft-cavage-http-signatures published
- Used by editors in web payments draft specifications (W3C)

2014+
- Used by a number of different separate efforts

2018
- Adopted by EU financial API standards groups (Polish API, Berlin Group)

2019
- Editors of draft publish major updates in draft -11
- Everyone using it freaks out and pegs to draft -10

2019+
- People realize that this isn't really a *standard* yet
- We figure out how to make this into a real standard?

What should we do?

- Combine drafts?
- Find a WG?
- Found a WG?