

# TACK (for pinning)

Trevor Perrin & Moxie Marlinspike

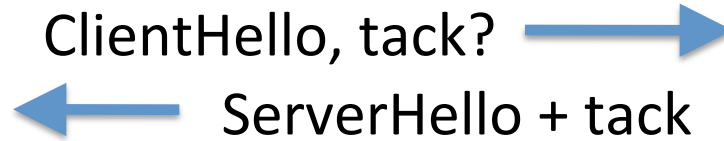
# Goals

- Pinning in TLS
  - Authenticate servers using “pins” from previous connections (“key continuity”)
  - Allow servers to “assert” pins
- Coexist with other server auth methods
  - As secondary method (e.g. HTTPS)
  - As primary method (e.g. SMTP between MTAs)
  - Usable for preloads, lookups, logs, etc...

# TACK Overview



**Client**



**Server**

**public\_key** : ECDSA-256  
**min\_generation** : 0-255  
**initial\_time** : UTC  
**end\_time** : UTC

pin

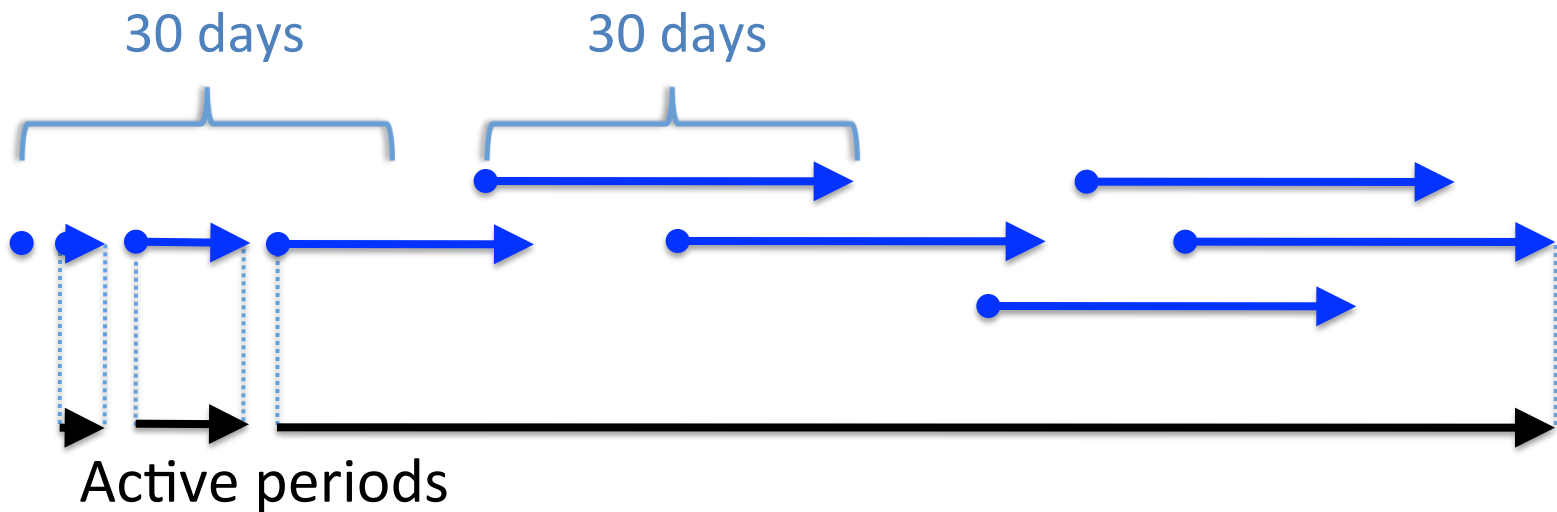
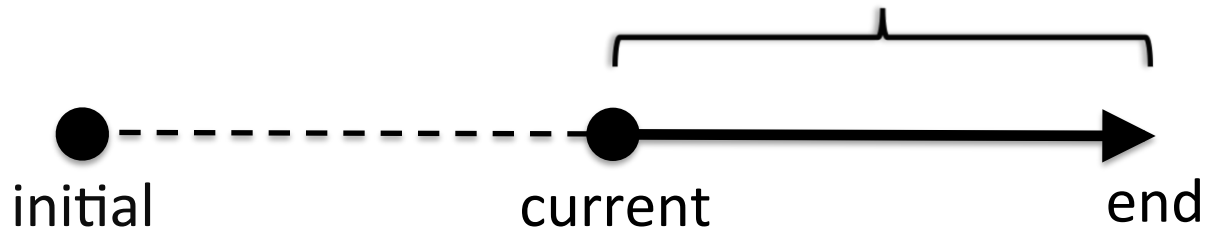
**public\_key** : ECDSA-256  
**min\_generation** : 0-255  
**generation** : 0-255  
**expiration** : minutes UTC  
**target\_hash** : SHA256  
**signature** : ECDSA-256

tack

(pins are created from tacks  
or received out-of-band)

# Pin Activation

Active period duration =  $\text{MIN}(30 \text{ days}, \text{current} - \text{initial})$



# Deployment

- Command-line tool
  - `tack genkey > KEY.pem`
  - `tack sign -k KEY.pem -c CERT.pem > TACK.pem`
- Apache
  - `SSLTACKTackFile: TACK.pem`
  - `SSLTACKActivationFlags: 1`

# Private key handling

- Password-protected file on laptop, smartcard, HSM...
- Can be deployed such that compromised/lost TACK key does not harm server availability
  - nonrevokable / nonexpiring tacks
- Make lots of backups
  - for safety, prefer compromised key to lost key
- Could outsource TACK private key

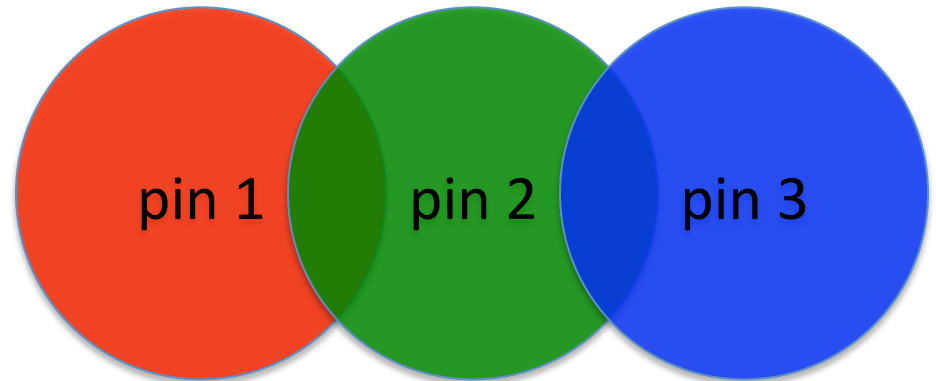
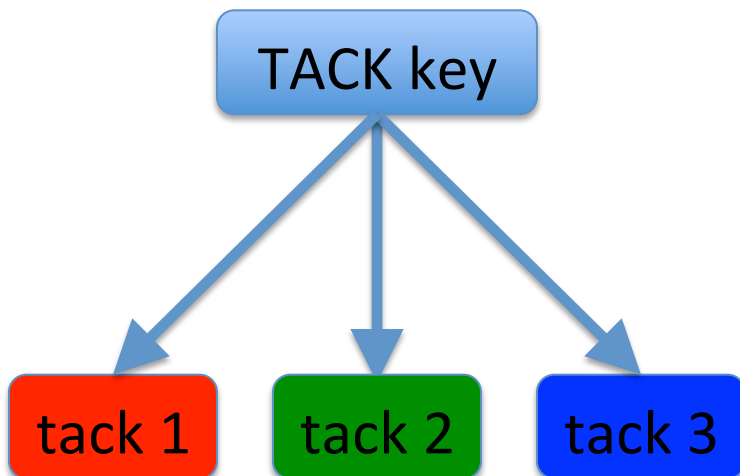
# Alternative: “shifting” pins

## Challenges with shifting pins

- pin activation
- network perspective
- overlapping pins
- other sig uses (break sigs etc?)

## Challenges with shifting “EE” pins

- complex private key mgmt
- multiple-key sites
- short-lived keys
- out-of-band pinning



Ex: (K1,K2,K3,K4) → (K3,K4,K5,K6) → ...

Shifting pins could use CA or EE keys

# Alternative: CA pins

- Choosing CAs and pinning correctly is complex
  - CAs may issue or alter cert chains unexpectedly
  - Clients may construct cert paths unexpectedly
- Likely less secure than pinning a TACK key
  - Multiple CAs; weak authentication of cert requests
- But easy if someone tells you what keys to pin and what CAs to use
  - Is that a good model?
  - Is key pinning the best way to achieve that?
  - Might some users prefer different models?



# Thanks!

- <http://tack.io>
  - Github (cmdline tool, server patches, client libs)
  - Test server
  - Mailing list