TLS Origin-Bound Certificates (TLS-OBC)

IETF 81 Presentation
Dirk Balfanz (Google)

Goal: Stronger Authentication for the Web

- Move away from **bearer tokens** on the web

- Instead, authenticate through **asymmetric cryptography**

- Long-term (with ubiquitous TPMs):
  malware can't remove credentials from host
- Short-term:
  render cookie theft useless (e.g., through XSS)

- Don't change things too much
  (keep cookies, keep existing datacenter architecture, etc.)

Why not use TLS Client Auth? Because it has problems:

- User Experience
  - Cert generation has UI
  - Cert selection has UI
    (happens before user can see content of web site)

- Privacy
  - user identity is same across all web sites

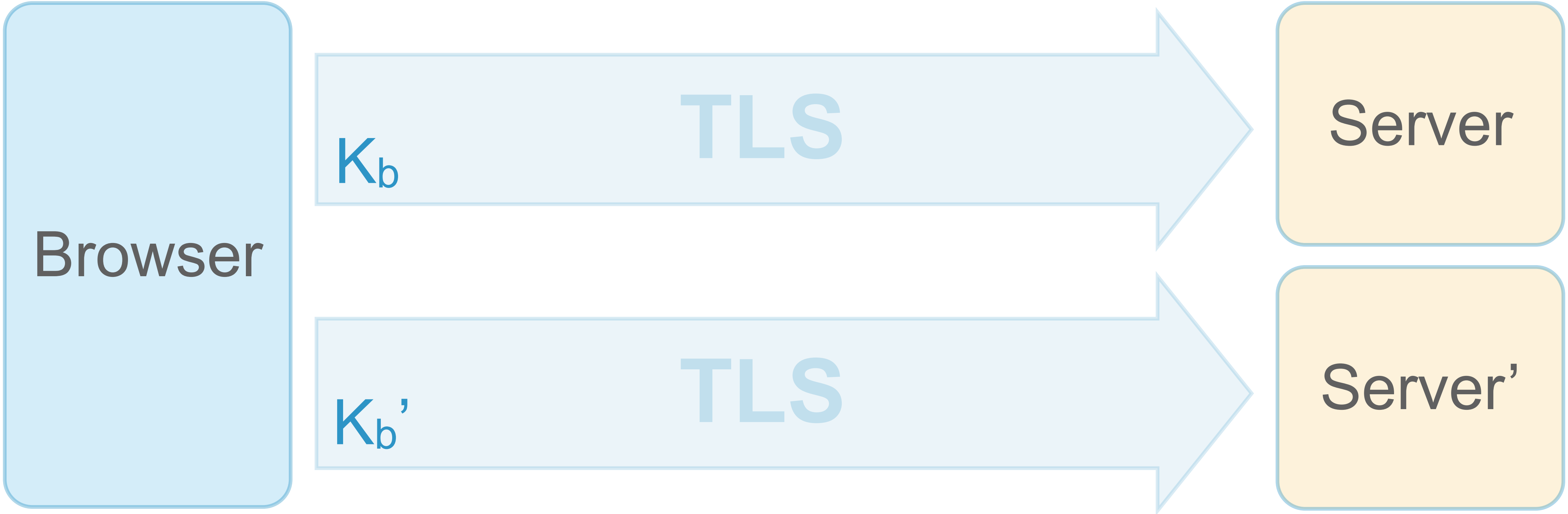
- Portability
  - moving certs is a hassle

- Problems in Datacenters
  - make TLS terminators part of the TCB

Origin-Bound Certs

- When asked by server, browser will:
  - create **self-signed cert** on the fly (**no UI**)
  - use it as TLS Client Auth cert with that server

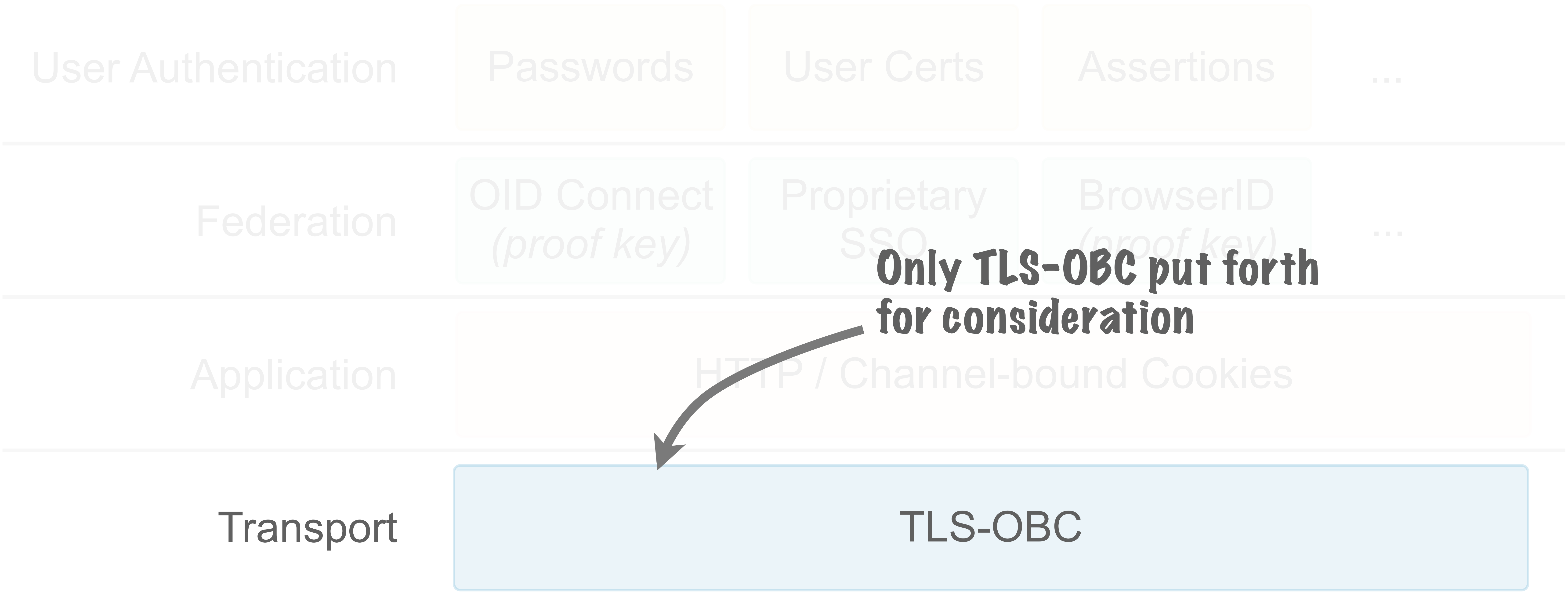
- Origin-bound certs are **like cookies**: They...
  - ...are per origin
  - ...are per browser profile
  - ...are ephemeral in incognito mode
  - ...can be cleaned out by the user



The Full Stack

User Authentication	Passwords	User Certs	Assertions	...
Federation	OID Connect *(proof key)*	Proprietary SSO	BrowserID *(proof key)*	...
Application	HTTP / Channel-bound Cookies			
Transport	TLS-OBC			

The Full Stack



User Authentication: Passwords | User Certs | Assertions | ...

Federation: OID Connect *(proof key)* | Proprietary SSO | BrowserID *(proof key)* | ...

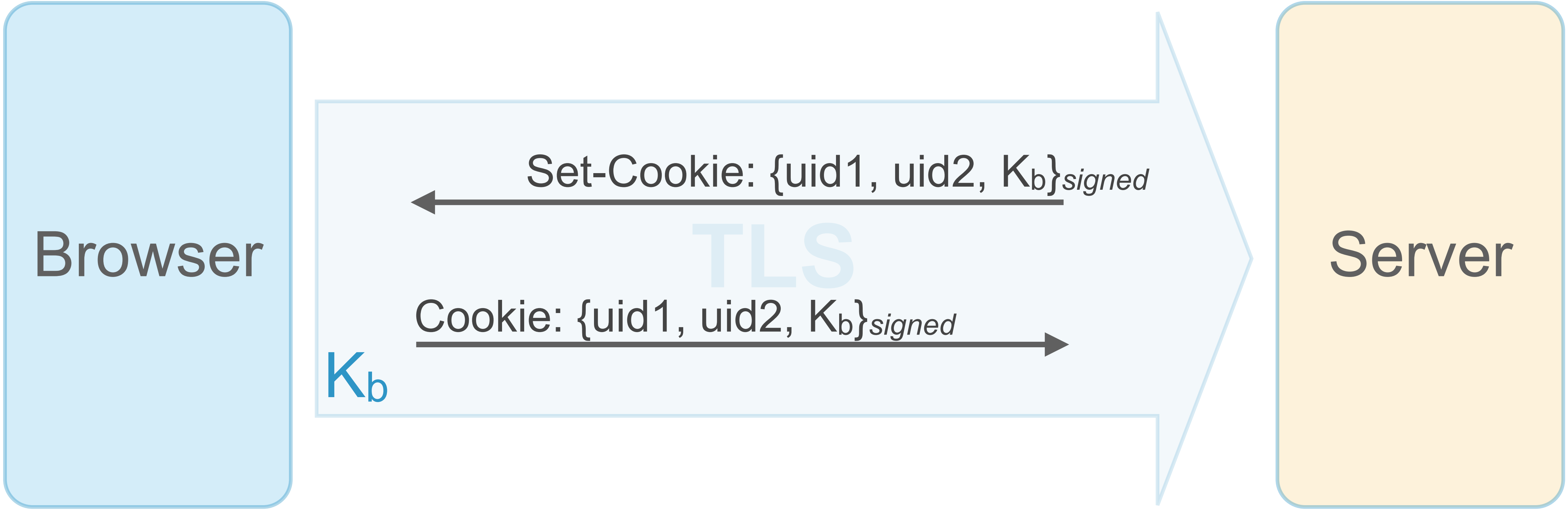
Application: HTTP / Channel-bound Cookies

Transport: TLS-OBC

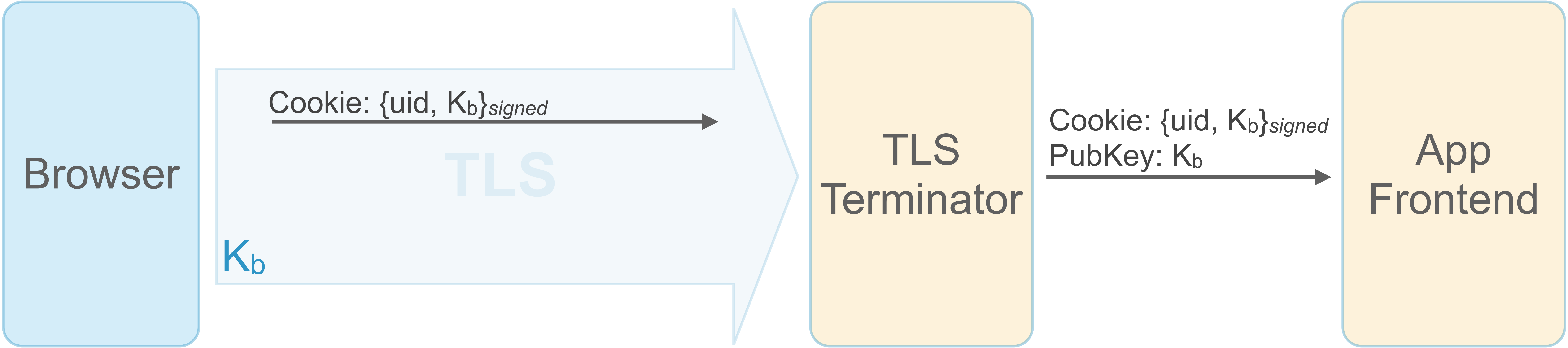
Only TLS-OBC put forth for consideration

Channel-Bound Cookies

- Servers can bind cookies to client certificate
  - client cert does not carry user-identifying information
  - login/logout as today: set/clear cookies
  - works with no login (unpersonalized), login, multi-login, over same session


Browser
Set-Cookie: {uid1, uid2, K_b}$_{signed}$
TLS
Cookie: {uid1, uid2, K_b}$_{signed}$
K_b
Server

TLS-OBC for Datacenters


Browser
Cookie: {uid, K_b}$_{signed}$
TLS
K_b
TLS Terminator
Cookie: {uid, K_b}$_{signed}$
PubKey: K_b
App Frontend

TLS-OBC Extension

- ServerHello/ClientHello negotiate extension
- Client generates origin-bound cert if necessary after server Certificate Request
- Client ignores issuers in server Certificate Request
  (should be set to empty by server)
- Client uses origin-bound cert normally as in Client Auth
- Server accepts self-signed certs, ignores not-before and not-after
- Client throws away cert, makes new keypair at own volition

- Should be used together with encrypted-client-cert extension to protect client privacy

Thanks!