OPAQUE IN TLS 1.3

NICK SULLIVAN
CLOUDFLARE
@GRITTYGREASE

H. Krawczyk, IBM Reasearch

R. Barnes, O. Friel, Cisco

N. Sullivan, Cloudflare

MODERN PASSWORD-BASED AUTHENTICATION IN TLS

A SHORT HISTORY

- Password-based authentication
  - * without sending the password to the server
- SRP (Secure Remote Password) – RFC 2945
  - aPAKE (Augmented Password-Authenticated Key Exchange)
  - Widely implemented, used in Apple iCloud, ProtonMail, etc.
- Dragonfly – RFC 8492
  - SPEKE (Simple password exponential key exchange) derived
  - Independent submission

SRP IN TLS (RFC 5054)

- Salt sent in the clear
  - Leads to pre-computation attack on password database
- Unsatisfying security analysis
- Finite fields only, no ECC
- Awkward fit for TLS 1.3
  - Needs missing messages (challenges outlined in draft-barnes-tls-pake)
  - Post-handshake requires renegotiation

OPAQUE

A new methodology for designing secure aPAKEs

OPAQUE OVERVIEW

- Methodology to combine an authenticated key exchange (such as TLS 1.3) with an OPRF (Oblivious Pseudo-Random Function) to get a Secure aPAKE
- Desirable properties
  - Security proof
  - Secure against pre-computation attacks
  - Efficient implementation based on ECC

OPAQUE DEPENDENCIES OVERVIEW

- Underlying cryptographic work in CFRG
  - OPAQUE (draft-krawczyk-cfrg-opaque)
  - OPRF (draft-sullivan-cfrg-voprf)
  - Hash-to-curve (draft-irtf-cfrg-hash-to-curve)

FUNDAMENTAL COMPONENTS / TERMINOLOGY

- The OPRF protocol allows the client to obtain a value based on the password and the server's private key without revealing the password to the server

- OPRF(pwd) is used to encrypt an envelope containing OPAQUE keys
  - The client's TLS 1.3-compatible private key
  - The server's TLS 1.3-compatible public key

FUNDAMENTAL COMPONENTS / TERMINOLOGY

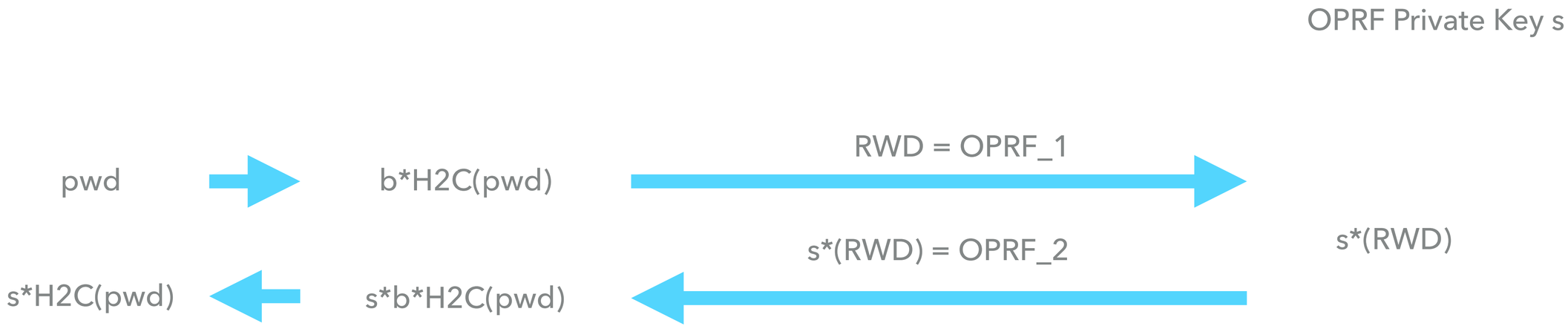
- Prime order group
  - e.g. The group of points on an Elliptic Curve such as P-256
  - Group elements will be denoted by capital letters such as P or Q
- Scalar multiplication
  - Adding a point to itself n times, such as P+P+…+P is denoted nP
  - Scalars will be represented by lower-case letters
- Hash to group element (H2C)
  - Function that takes a scalar and outputs a random group element

FUNDAMENTAL COMPONENTS / TERMINOLOGY

- OPRF Flow

FUNDAMENTAL COMPONENTS / TERMINOLOGY

- OPRF Flow



FUNDAMENTAL COMPONENTS / TERMINOLOGY

- The OPRF protocol allows the client to obtain a value based on the password and the server's private key without revealing the password to the server

- OPRF(pwd) is used to encrypt an envelope containing OPAQUE keys
  - The client's TLS 1.3-compatible private key
  - The server's TLS 1.3-compatible public key

FUNDAMENTAL COMPONENTS / TERMINOLOGY

- OPRF Flow

HIGH-LEVEL OVERVIEW

- User creates the envelope during password registration by running OPRF
- User proves knowledge of the password by being able to open the envelope and use the OPAQUE private key inside
- OPAQUE private keys used to authenticate handshake
    1. In place of PKI keys with a new Certificate Type
    2. Combined with TLS ephemerals in the key schedule using MQV or 3DH

IN PLACE OF PKI KEYS: OPAQUE-SIGN

- OPAQUE keys are signature keys
- Client sends identity, OPRF_1
- Server
  - Certificate message with OPRF_2 in extension
  - CertificateRequest with Identity
- Key used for server CertificateVerify is server OPAQUE key
- Key used for client CertificateVerify is client OPAQUE key

IN KEY SCHEDULE: OPAQUE-(3DH|MQV)

- OPAQUE keys are TLS 1.3 key shares
- Client sends identity, OPRF_1, key_share matching OPAQUE key type
- Server
  - EncryptedExtensions message with OPRF_2 in extension
- Ephemeral-Ephemeral-Static-Static key exchange used as input to key schedule
  - K -> HKDF-Extract = Master Secret
- Optional certificate auth

POST-HANDSHAKE AUTH: OPAQUE-SIGN IN EXPORTED AUTHENTICATOR

- Client to Server
  - Authenticator Request: Identity, OPRF_1
- Server to Client
  - Exported Authenticator: OPRF_2, Signature
  - Authenticator Request: Identity (linked to previous EA)
- Client to Server
  - Exported Authenticator: Signature

PROPERTIES

- OPAQUE-Sign
  - No username privacy without ESNI-like mechanism
  - No simultaneous PKI auth
- OPAQUE-3DH, OPAQUE-MQV
  - No username privacy without ESNI-like mechanism
  - Optional PKI auth
- OPAQUE-Sign in Exported Authenticators
  - Username privacy
  - Optional PKI auth
  - Post-handshake auth through HTTP/2-like mechanisms
  - Extra round-trip

RECAP

- New password-based authentication mechanism for TLS 1.3
- First Secure aPAKE protocol provably secure against pre-computation attacks
- Multiple constructions with desirable properties

- Interesting for the WG to pursue as an alternative to SRP?

OPAQUE IN TLS 1.3

NICK SULLIVAN
CLOUDFLARE
@GRITTYGREASE

H. Krawczyk, IBM Reasearch

R. Barnes, O. Friel, Cisco

N. Sullivan, Cloudflare