OPAQUE:

OPRF-based asymmetric* PAKE

* a.k.a. augmented

[S. Jarecki, H. Krawczyk, J. Xu, Eurocrypt 2018]

[draft-krawczyk-cfrg-opaque-00]

aPAKE: ‘a’ for asymmetric/augmented

- Password-Authenticated Key Exchange in the <u>client-server setting</u>
  - **aPAKE requirements**: PKI free and security against server compromise (forces offline dict attack) prevent *pre-computation attacks*
  - In other words, best possible security, only unavoidable attacks allowed: online guesses + offline upon server compromise
- Compare password-over-TLS:
  - Prevents pre-computation (via salted hashes) but fully dependent on PKI + server sees passwd (and so do middle boxes, termination points, MitM, etc.)
- Clearly, aPAKE is better (no PKI dependence, server does not see pwd) … but is it, really?

All knonwn aPAKE protocols are vulnerable to pre-computation attacks!

- Why? *They do not accommodate secret salt*
  - Either they do not use salt at all or send it *in the clear* from server to user
- Wait, but there are aPAKE that are proven secure…
  - … Yes, but the standard aPAKE definitions do not exclude pre-computation attacks (this includes BMP'00 and GMR'06)
- Worse than password-over-TLS in this *fundamental* aPAKE aspect

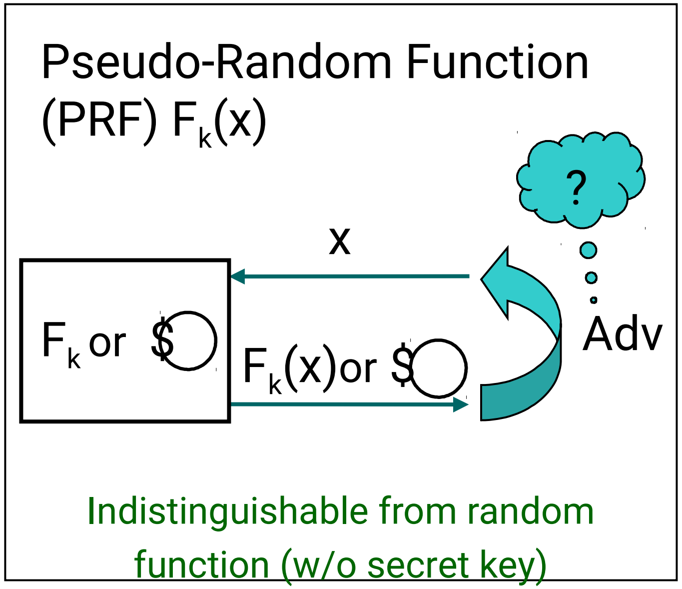
This includes SRP, SPAKE2+, AugPAKE, VTBPEKE, etc.

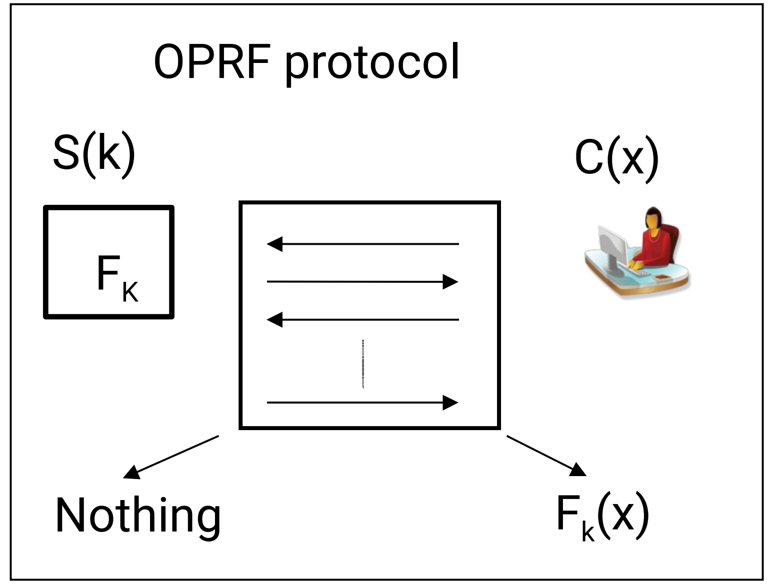
Is this essential (proven impossibility)?

Nope…

OPAQUE: First aPAKE secure against pre-computation (and with proof)

Oblivious PRF (OPRF)





- OPRF: An interactive PRF “service” that returns PRF results *without the server learning the input or output of the function*

OPAQUE: Basic idea

Follows FK'00, Boyen'09, JKKX17

- Assume KE protocol w/ private-public keys $priv_U$, pub_U, $priv_S$, pub_S
- Define $\boxed{rwd = OPRF_K(pwd)}$; U has pwd, S has K, only U learns rwd
- Server stores $C = AuthEnc_{rwd}(priv_U, pub_S)$, $priv_S$ and OPRF key K
- For login:
  - U and S run OPRF protocol, so U obtains rwd
  - S sends C to U, so U obtains $priv_U$, pub_S
  - U and S run KE with keys ($priv_U$, pub_U, $priv_S$, pub_S)
- A "compiler" from any KE to an aPAKE (with any OPRF) .

  -modular and flexible

DH-OPRF

- **PRF**: $f_k(x) = H(x, v, H'(x)^k)$ over group G with generator g; k is a key, $v = g^k$; H' hashes x into a random element in G.
- Oblivious computation via Blind DH Computation (C has x, S has k)
  - **C, on input x, chooses random** r, sends $a = g^r \cdot H'(x)$ to S
  - **S replies with** $b = a^k$ and v
  - **C sets** $c = b \cdot v^{-r}$ and $f_k(x) = H(x, v, c)$
    - Note that $c = b \cdot v^{-r} = (g^r \cdot H'(x))^k \cdot (g^k)^{-r} = (H'(x))^k$

Server: 1 var-base exponent'n
Client: 1 var-base, 1 fixd-base
Single round

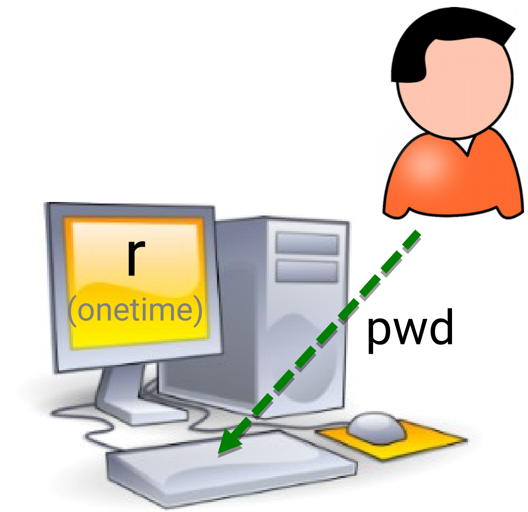
- The blinding factor g^r works as a one-time encryption key, hence ***it hides $H'(k)$ and x perfectly*** (from S).

OPAQUE with DH-OPRF

server stores C_U = $\mathrm{AuthEnc}_{rwd}(priv_U, pub_S), priv_S, K_U, v_U = g^{K_u}$

$U, a = H'(pwd) \cdot g^r, g^x$

$C_U, v_U, b = a^{K_u}, g^y$



$SK = KE(priv_S, y, pub_U, g^x)$

$rwd = H(pwd, v_U, b \cdot v_U^{-r})$

$priv_U, pub_S = \mathrm{Dec}_{rwd}(C_U)$

$SK = KE(priv_U, x, pub_S, g^y)$

- E.g., KE=HMQV. total # expon's (fixed base/ variable base):

  Client 2 fixed base, 2.17 var base, Server 1 fixed base, 2.17 var base

OPAQUE Performance

- Single round w/ implicit authentication + 1 msg for explicit auth'n
- Cost: KE + 1 server exponentiation, 2 client exponentiations*

  * One or two fixed-base exponentiations (g^r, v^{-r}) for user

- OPAQUE with HMQV (# exp's): Client 2 fixed base, 2.17 var base, Server 1 fixed base, 2.17 var base (about 2.5 exp each)
  - Similar to SPAKE2+ in performance
  - but with security against pre-computation and with a proof
  - and flexibility for choice of KE (e.g HMQV*, SIGMA, TLS, etc.)

* HMQV patent: may be solvable if real interest in standardizing

OPAQUE with TLS 1.3

- *Reuse* DH exchange of TLS DH exchange, use $priv_U$ as signature key for client authentication (perfect fit with 3-flight handshake)
- User account privacy: use resumption key if available
  Or: Add extra round trip (between TLS 2nd and 3rd flight)
  - post-handshake client auth'n and exported authenticators may help

OPAQUE Security

- Secure against pre-computation attacks (secret salt)!!
- Proof
  - Strong aPAKE model (PKI-free and disallows pre-computation attacks)
  - Proof of OPAQUE is generic: OPRF + KE (with KCI)
  - With DH-OPRF: In ROM under Gap-OMDH
- Forward security
- User-side hash iterations
  - increased security against offline attacks upon server compromise

OPAQUE Features

- **Efficient, provably secure** and …
- **No reliance on PKI**
- Server **never sees password**, *not even at init* (good against pwd reuse)
- **Private salt**: Attacker cannot pre-compute dictionary
- Hash **iterations can be offloaded** to user
- **TLS integration** (hedged PKI: PAKE-protected TLS)
- **Storing other user secrets**
- User-transparent server-side **threshold implementation**

Final Remarks

- IF we are looking for a strong aPAKE to standardize (are we?) OPAQUE seems to fit perfectly
- In particular, a good fit for TLS 1.3
- Passwords are not going away, so let's improve their use
  - Additional new tools help too: Sphinx password manager, TOPPSS password protected secret sharing, …