Strengthening Master Secrets (to get good TLS Channel Identifiers)

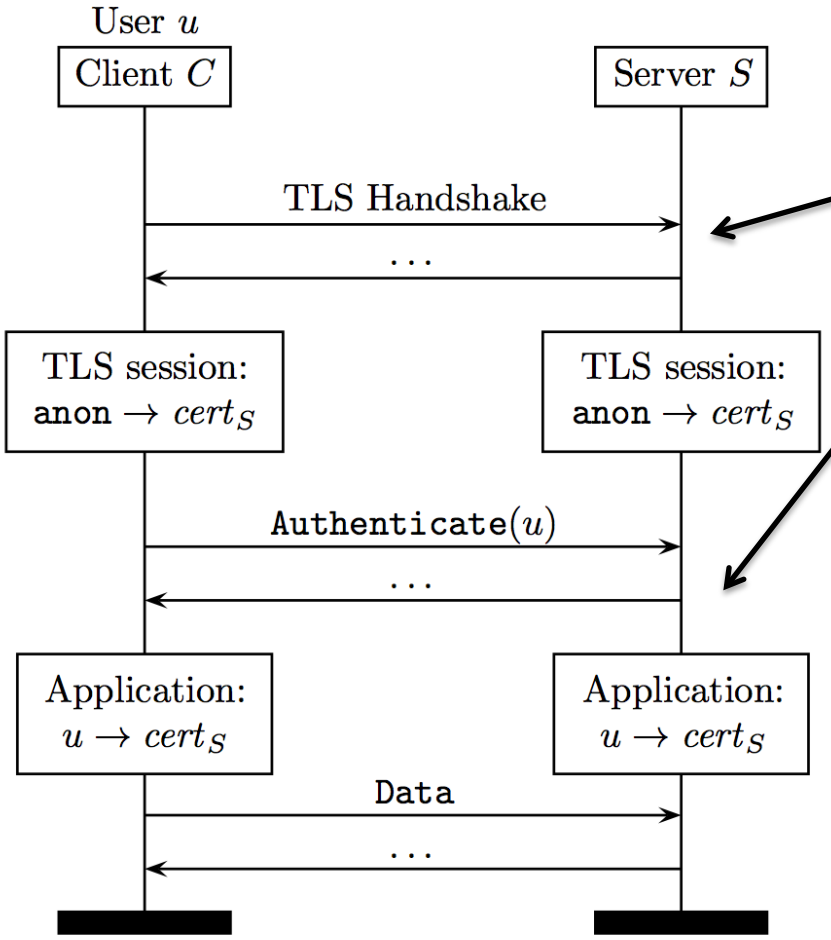
K Bhargavan, A Delignat-Lavaud, A Pironti (INRIA)
A Langley (Google), M Ray (Microsoft)

https://secure-resumption.com

IETF'89, London, March 4, 2014

Joint work with C. Fournet, P-Y Strub, M. Kohlweiss, S Zanella-Béguelin

Authentication over TLS



- Common Pattern
  - *Outer*: server-authenticated TLS
  - *Inner*: user authentication protocol
- Many examples
  - SASL, GSSAPI, PEAP, …
  - Renegotiation with client certificate
- Common concerns
  - *How to bind inner authentication with outer channel?*

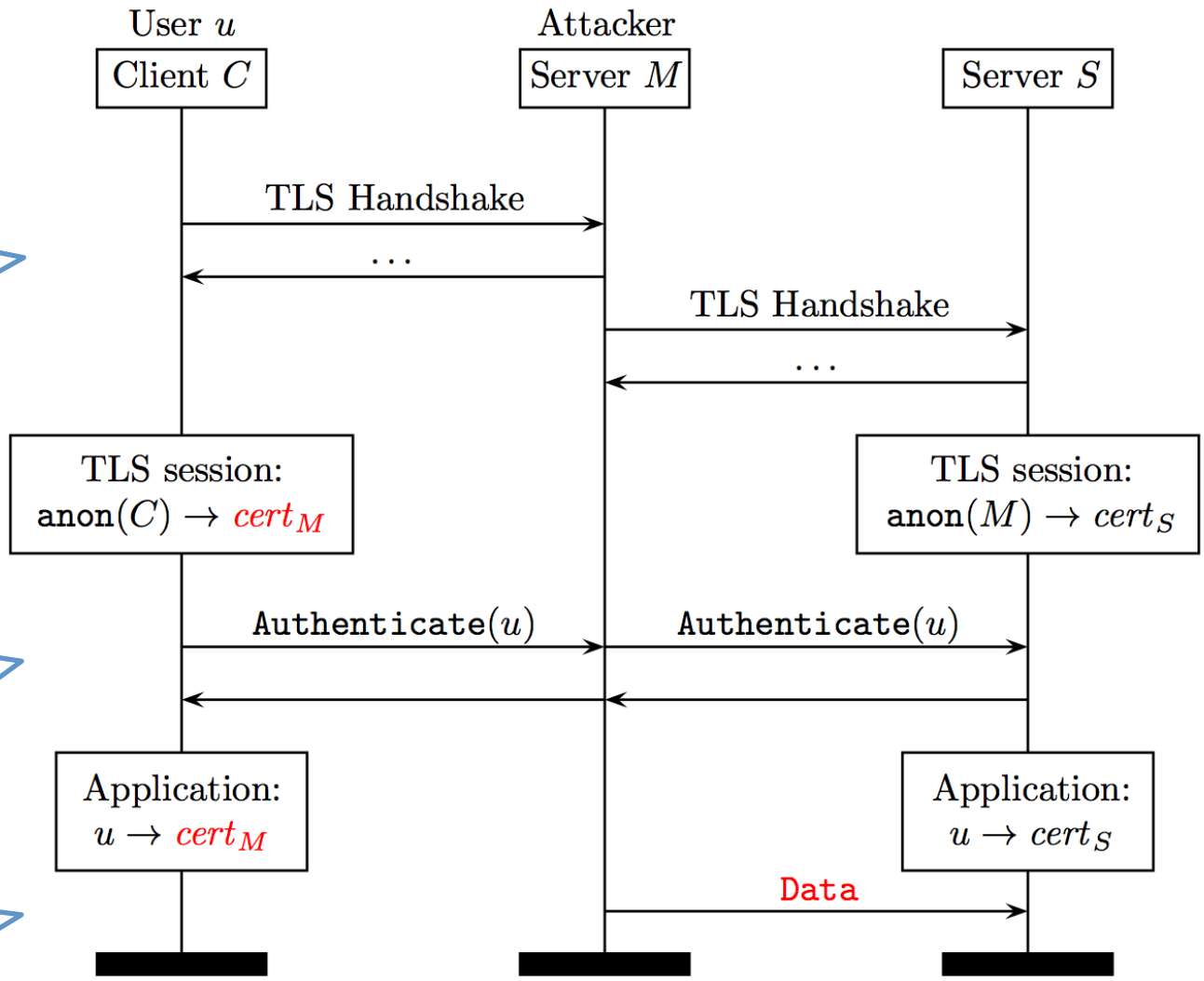
MITM on Tunneled Auth [2002]

Simplified version of [Asokan, Niemi, Nyberg'02]

User *u* connects to malicious server *M*

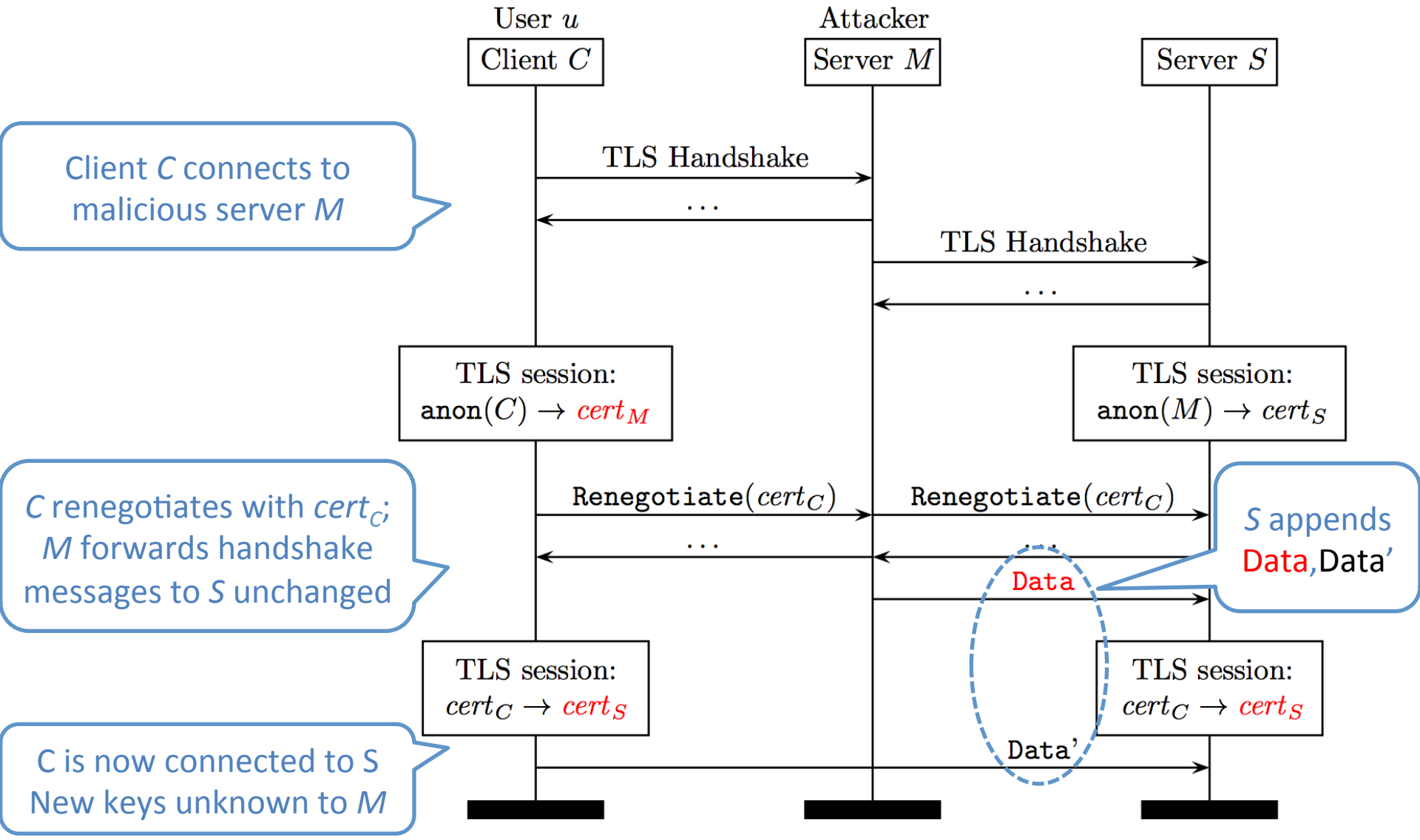
M forwards user auth messages to *S* unchanged

M can send Data to S pretending to be *u*



TLS Renegotiation [2009]

Martin Rex's Version



Attack Preconditions and Caveats

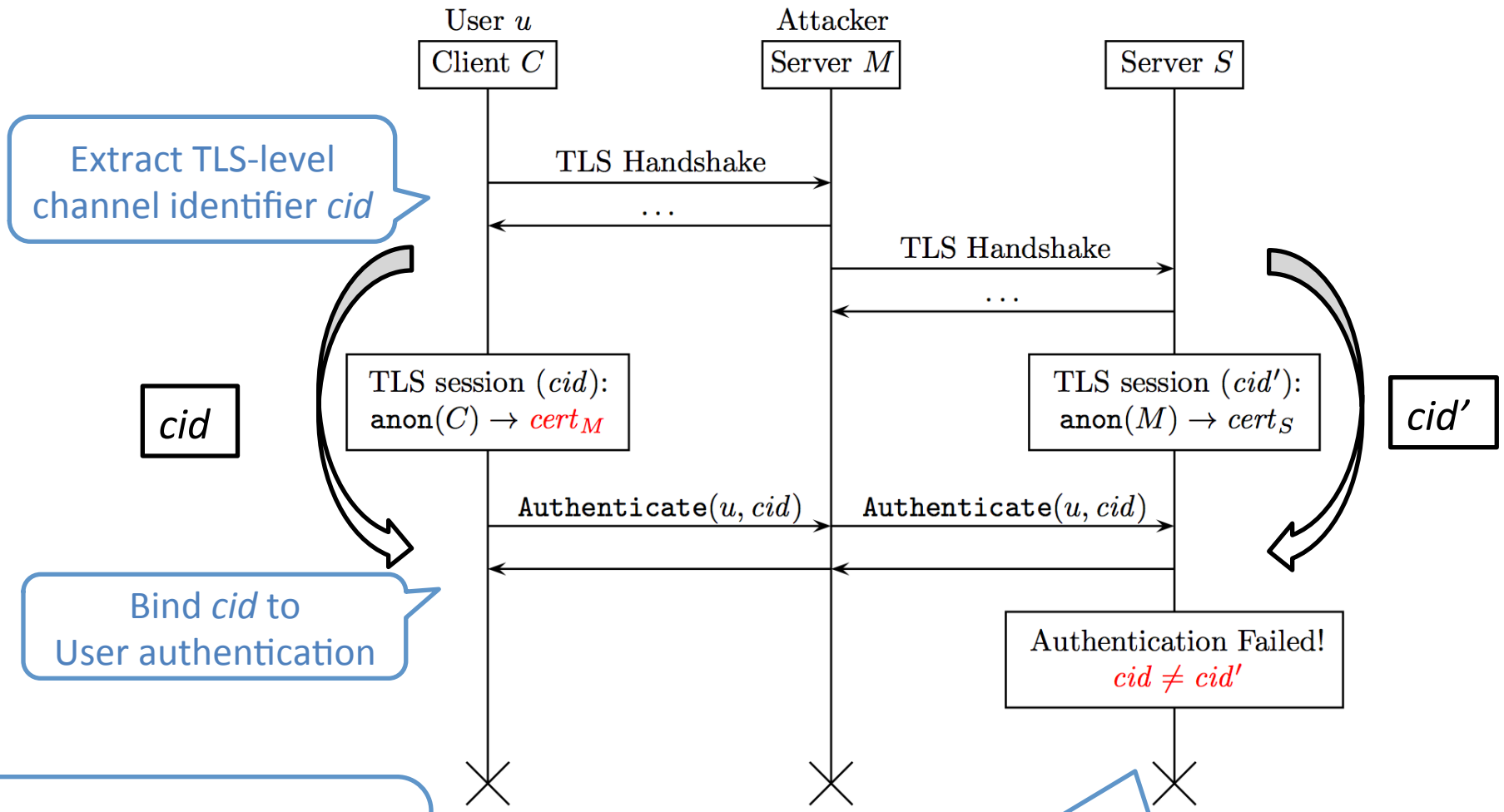
- The user *u* must be willing to authenticate to *M*
  - If TLS client refuses to connect,
    or user refuses to authenticate, they are safe

- For Rex's renegotiation attack
  - *C* is willing to accept a change of server cert from *M* to *S*
  - *S* concatenates data received before and after renego
  - If TLS client/server prevents these, they are safe

- In practice, all these preconditions are widely met.

  So, user authentication protocols implement
  various "channel binding" countermeasures.

Binding User Auth to TLS Channels



Choices for identifier:

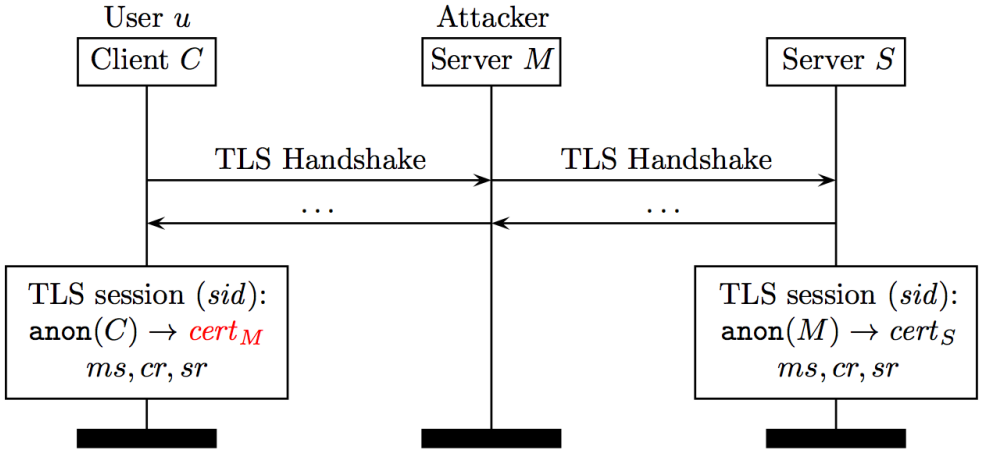
- ***cid=F(master_secret)***
- ***cid=F(verify_data)***

Does not work if *M* can ensure that cid = cid'

Triple Handshake Attack: 1

- In the RSA handshake, *M* can ensure that the master secrets on both connections is the same
  - *M* re-encrypts *C's* premaster secret under *S*'s public key
  - Uses same client and server randoms on both handshakes

- *M* can also do this with DHE handshakes
  - It chooses a "bad" Diffie-Hellman group

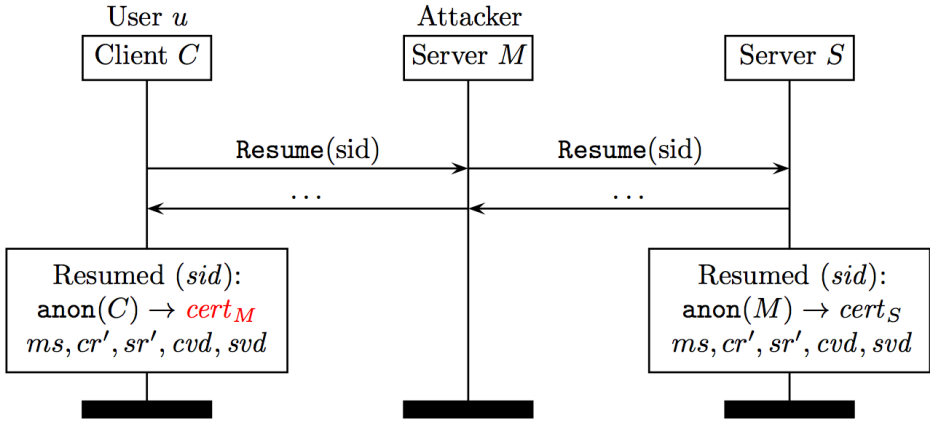


- *Impact*: The master secret is not a good channel identifier
- *Breaks*: Compound authentication (reenables 2002 attack)

Detailed message traces: https://secure-resumption.com

Triple Handshake Attack: 2

- If *C* resumes its session on a new connection, *M* can forward the abbreviated handshake to *S*
  - Works because master secrets, ciphersuites, session ID the same
  - Both new connections will have the ***same handshake log*** and ***same verify_data***
  - On both connections, same tls-unique=server_verify_data

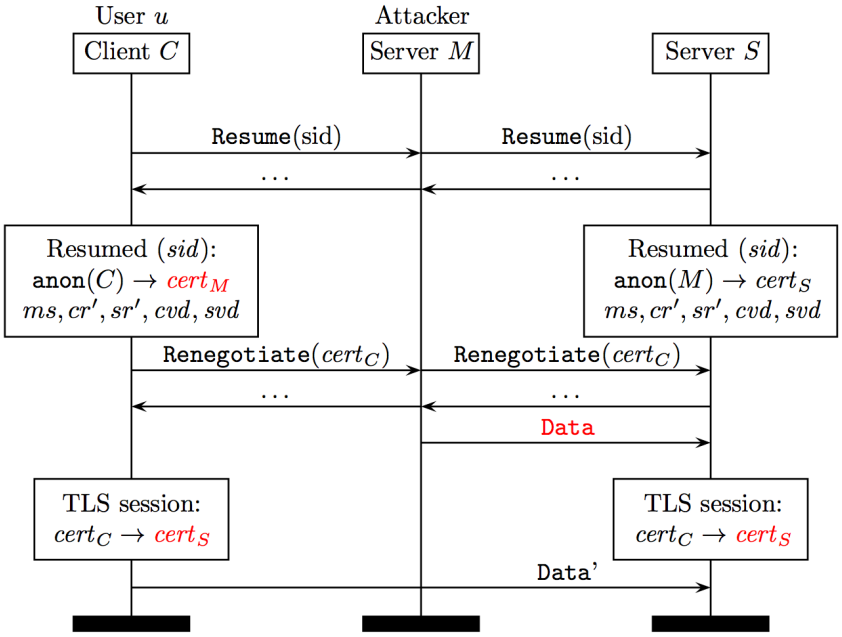


- *Impact*: tls-unique is not a good identifier after resumption
- *Breaks*: Channel Binding in SASL (SCRAM, GS2)

Triple Handshake Attack: 3

- If *C* renegotiates with client certificate;
  M can forward the renegotiation handshake to *S*
  - Works because renegotiation indication is the same

    RI = client_verify_data + server_verify_data



- *Impact*: Renegotiation Indication is not a good channel identifier after session resumption
- *Breaks*: Renegotiation Indication (reenable's Rex's attack)

Countermeasures

- For renegotiation, implementation checks will be enough
  - Always validate server certificate during renegotiation
  - Many TLS clients still need to do this
- For tls-unique, compound auth, no easy fixes
  - Don't rely on tls-unique after resumption
  - Ensure that clients only present user credentials to trusted servers

- *Root problem*: The master secret is not bound to the security context that created it
  - E.g. does not depend on client + server certificates

If we make the master secret a good session identifier, tls-unique and renegotiation indication will be fixed too!

Proposal: Extended Master Secret

- Compute a session hash for full handshakes

  session_hash = Hash(handshake_messages)

  - All messages up to and including ClientKeyExchange (master_secret will be needed in SSL 3.0 CertificateVerify)

- Add session hash to master secret derivation:

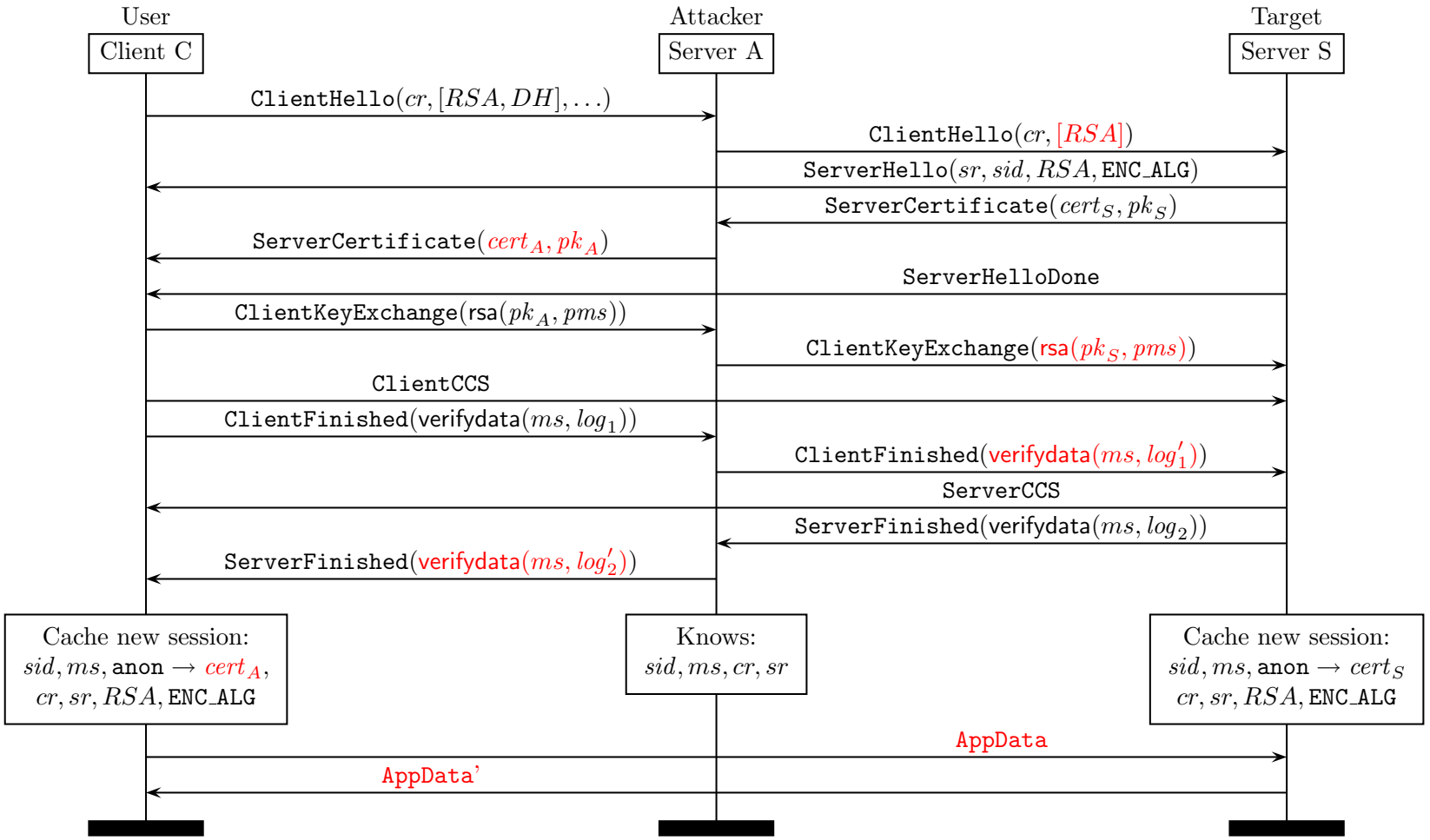
  master_secret = PRF(pre_master_secret,
  "extended master secret”,
  session_hash) [0..47];

- Extension draft: draft-bhargavan-tls-session-hash-00.txt
- Alternative proposal: fix resumption, à la RFC5746

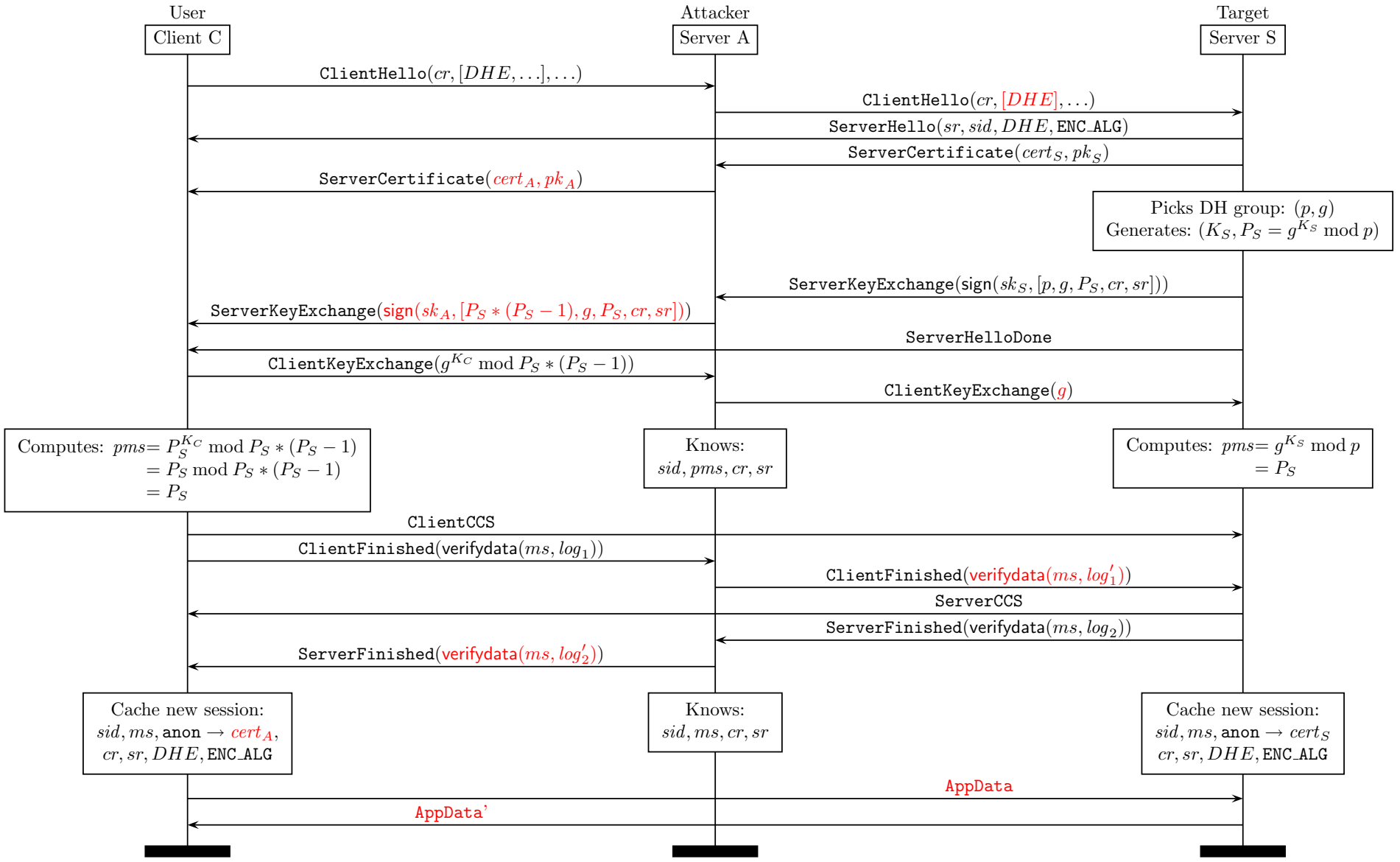
Questions?

- More details and research paper at
  - https://secure-resumption.com

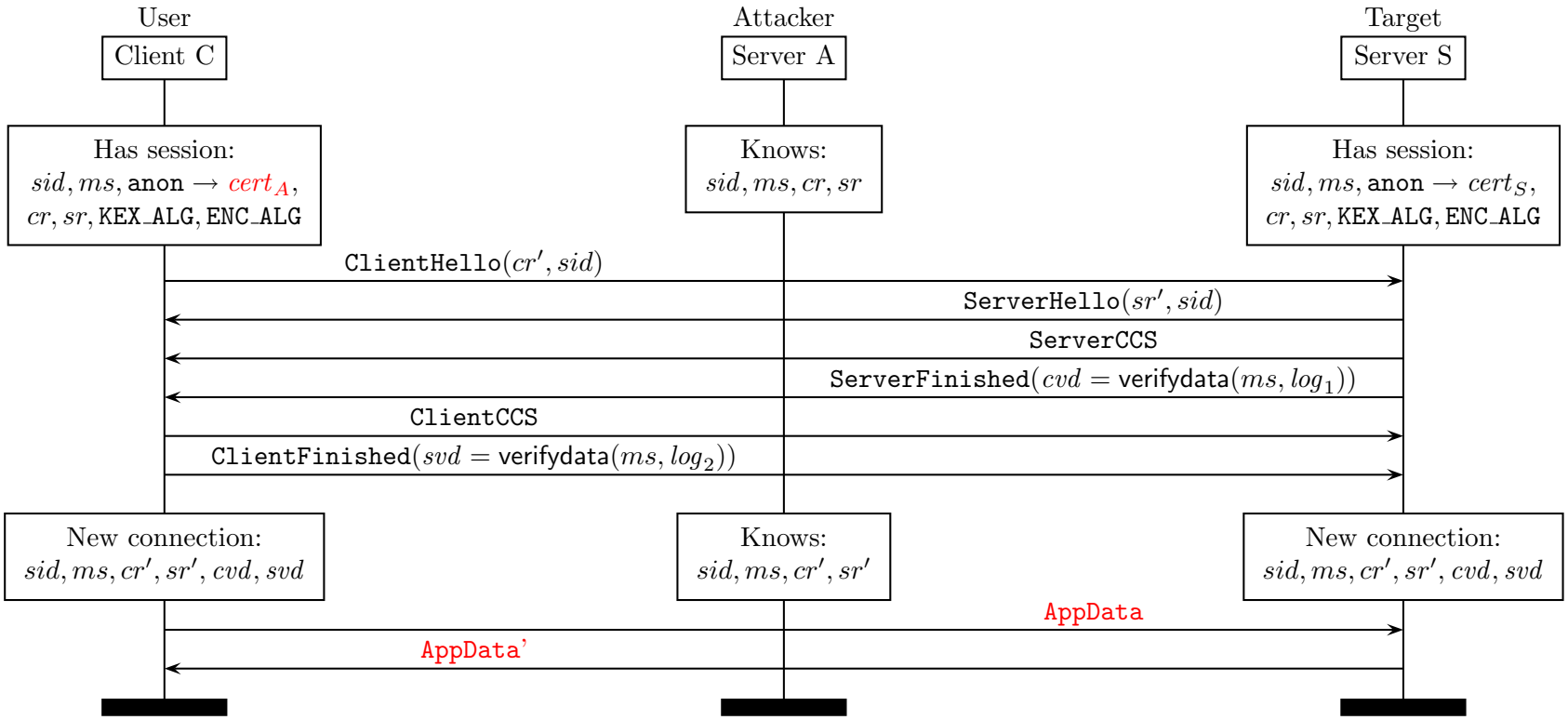
3HS Attack – step 1 (RSA)



3HS Attack – step 1 (DHE)



3HS Attack – step 2 (resumption)



3HS Attack – step 3 (renegotiation)