Verified Contributive Channel Bindings for Compound Authentication

Antoine Delignat-Lavaud, Inria Paris

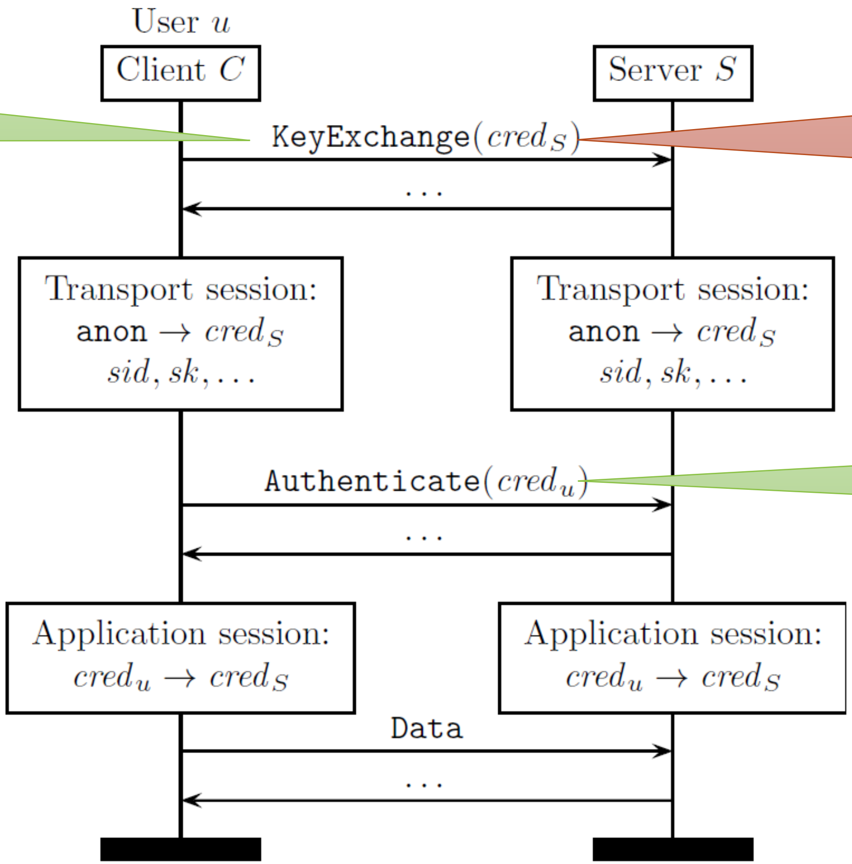
Joint work with Karthikeyan Bhargavan and Alfredo Pironti

Motivation: Authentication Composition

- Protocols for authenticated key exchange (AKE) and user authentication (UA) are well-studied and verified in isolation
- In practice, applications use complex sequences of AKE and UA protocols with re-keying, resumption and re-authentication

Motivation: Authentication Composition

TLS, IPsec, QUIC…



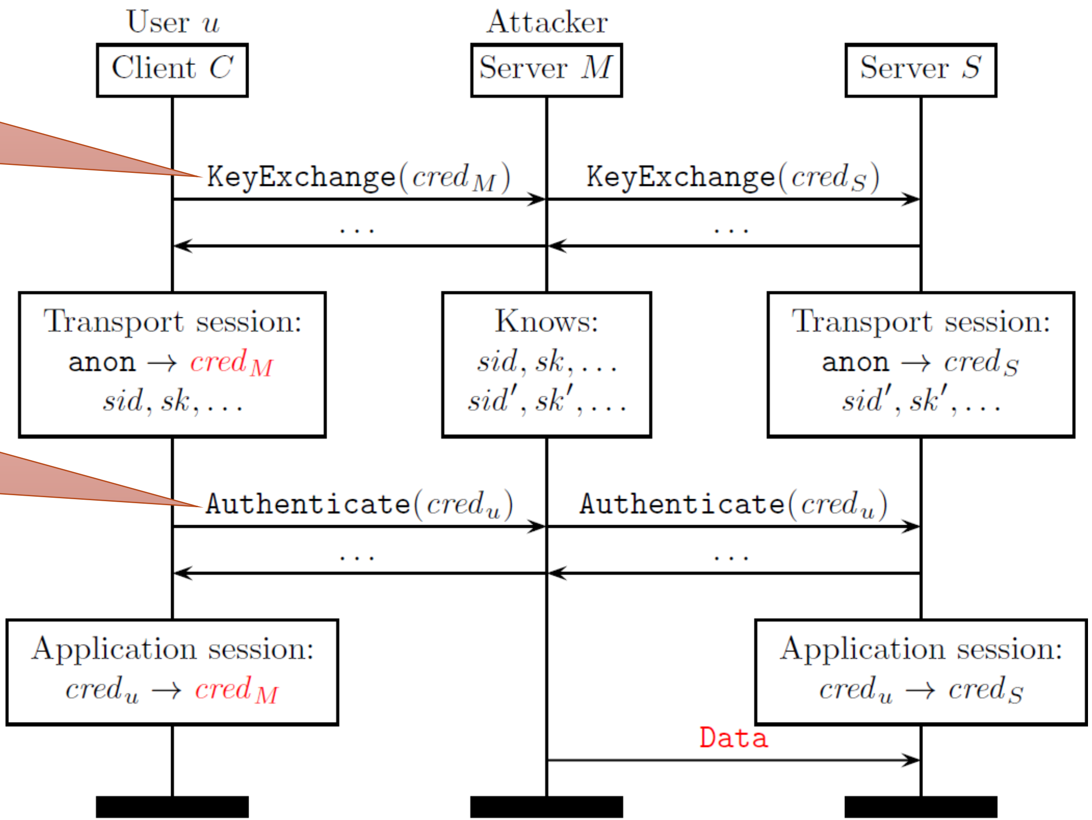
“Anti-pervasive” encryption: $cred_S = anon$

X.509, token, password…

Problem: Credentials Forwarding Attacks

C believes M=S

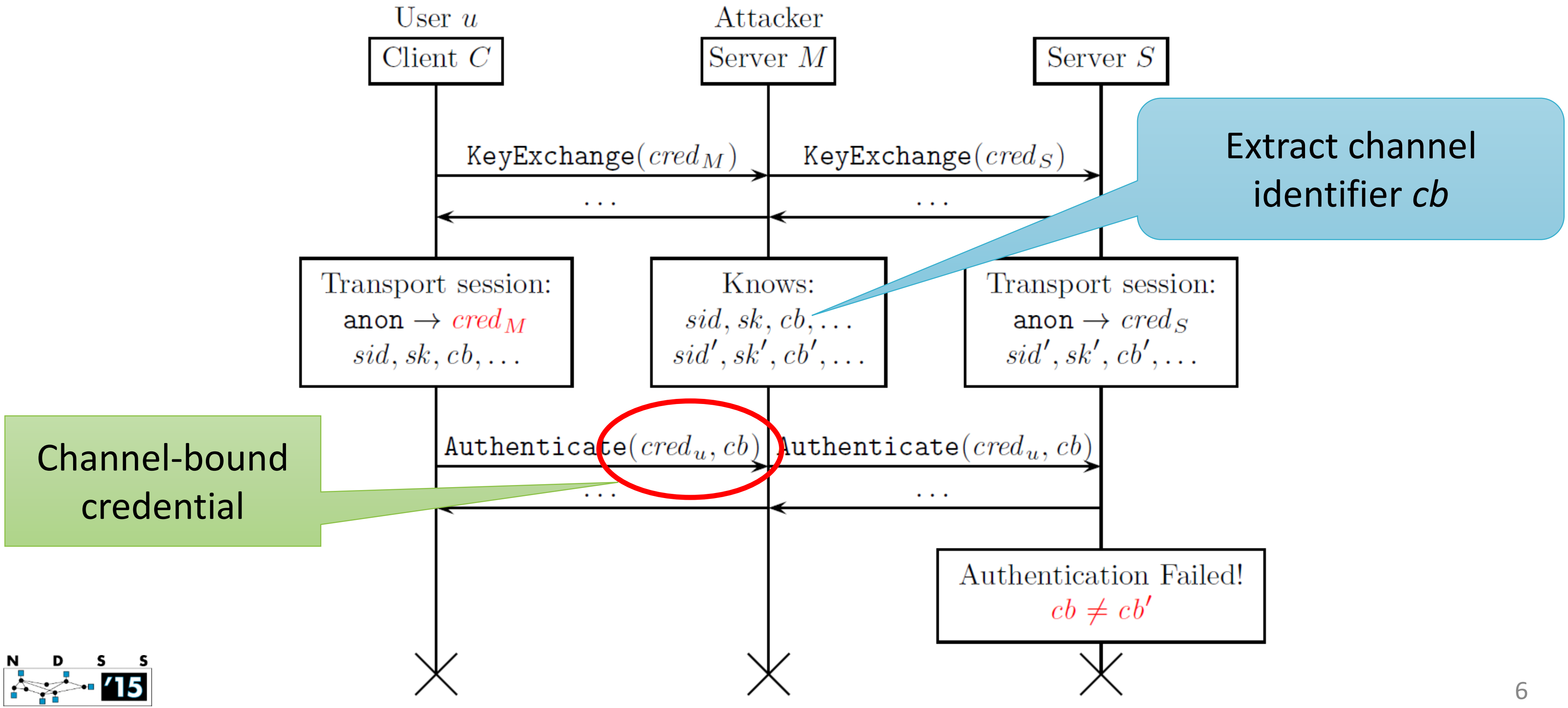
U uses same credential on M and S



Credentials Compromise is Real

- Password reuse, token leakage, etc.
- Key compromise (e.g. Heartbleed), PKI failure
- Validation failure
  - Certificate parsing (e.g. CVE-2014-1568 universal PKCS#1 forgery in NSS)
  - Protocol implementation bugs
    - Goto fail
    - Coming to Oakland: State Machine AttaCKs against TLS (smacktls.com)
  - User ignores warning
  - Application skips basic checks (e.g. host validation)

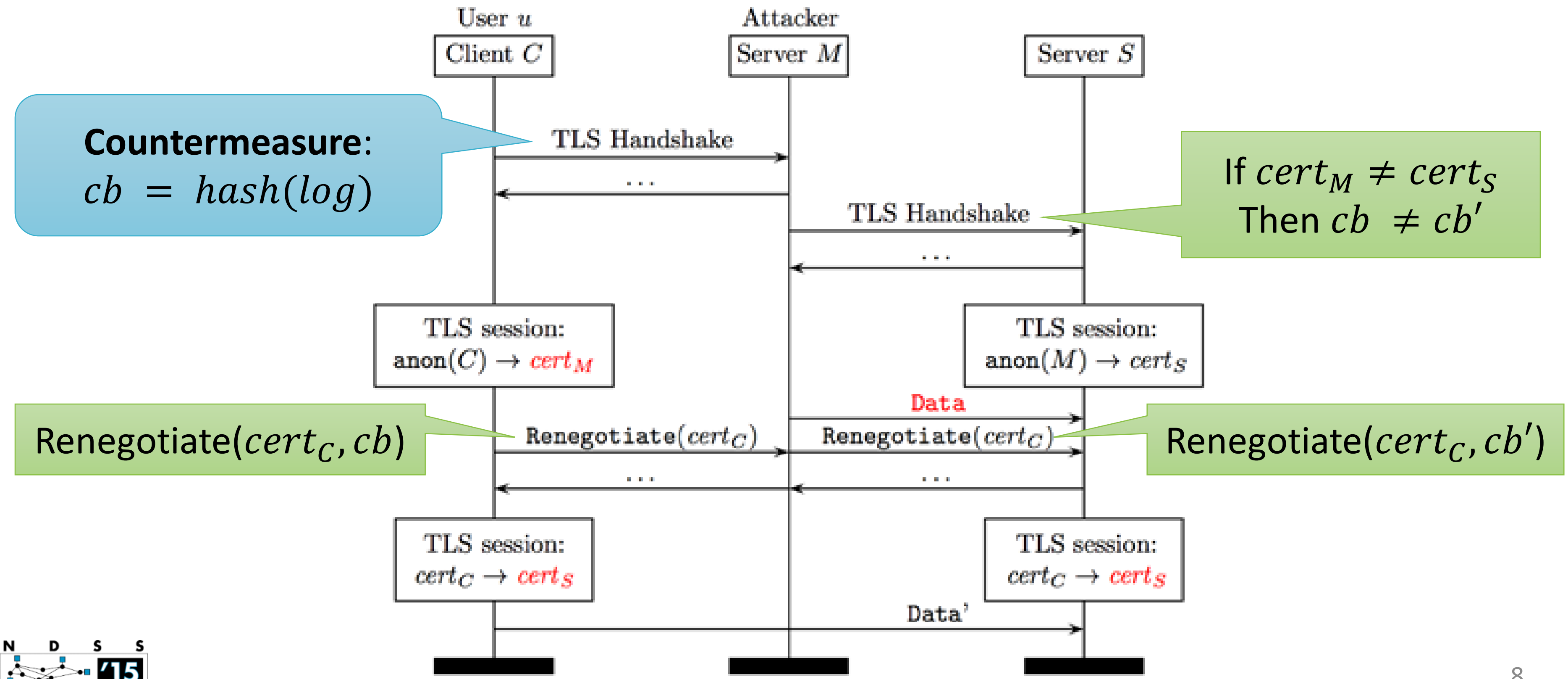
Typical Countermeasure: Channel Binding



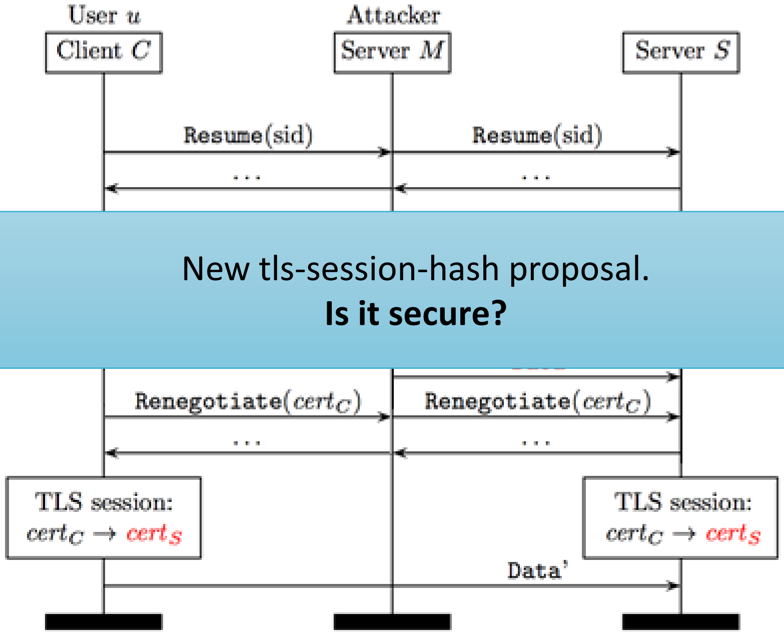
Examples of Typical Compound Protocols

Composed with TLS	Composed with IKEv2	Composed with SSH
EAP	EAP	USERAUTH
SASL (e.g. SCRAM-PLUS)	Re-authentication	Re-keying
Channel ID (e.g. cookie)	Resumption	
Renegotiation / Resumption		

TLS Renegotiation Attack [Ray & Rex 09]



Triple Handshake Attack [IEEE S&P'14]



New tls-session-hash proposal.
Is it secure?

$cb = (cvd, svd)$ doesn't prevent credential forwarding!

Research Questions

- What are the security goals of compound protocols?
- Which channel bindings effectively achieve these goals?
  - We want formal guarantees this time!

Threat Model

- Symbolic Dolev-Yao Attacker
  - Perfect cryptographic primitives
  - Attacker can freely instantiate any protocol with peers or act as MitM

- Credentials compromise
  - Client and server credentials can be compromised
  - Honest participants may be using compromised credentials

Formal Problem Statement

Definition: Agreement at <u>a</u> in Authentication Protocols

If:

- Principal <u>a</u> completes protocol instance *I*
- Peer <u>b</u> sent a non-compromised credential
- Session secrets in *I* have not been leaked

Then:

- <u>b</u> is not the attacker
- The dual instance *I'* ran by <u>b</u> agrees with *I* on public parameters and session secrets

Credentials *ci*, *cr*
Session identifier *sid*
Channel binding *cb*

Session key *sk*

Formal Problem Statement

Definition: Compound Authentication
A set of protocols $\{P_1, \ldots, P_k\}$ achieves compound authentication if, for any sequence of instances of these protocols, the following property holds:

If:

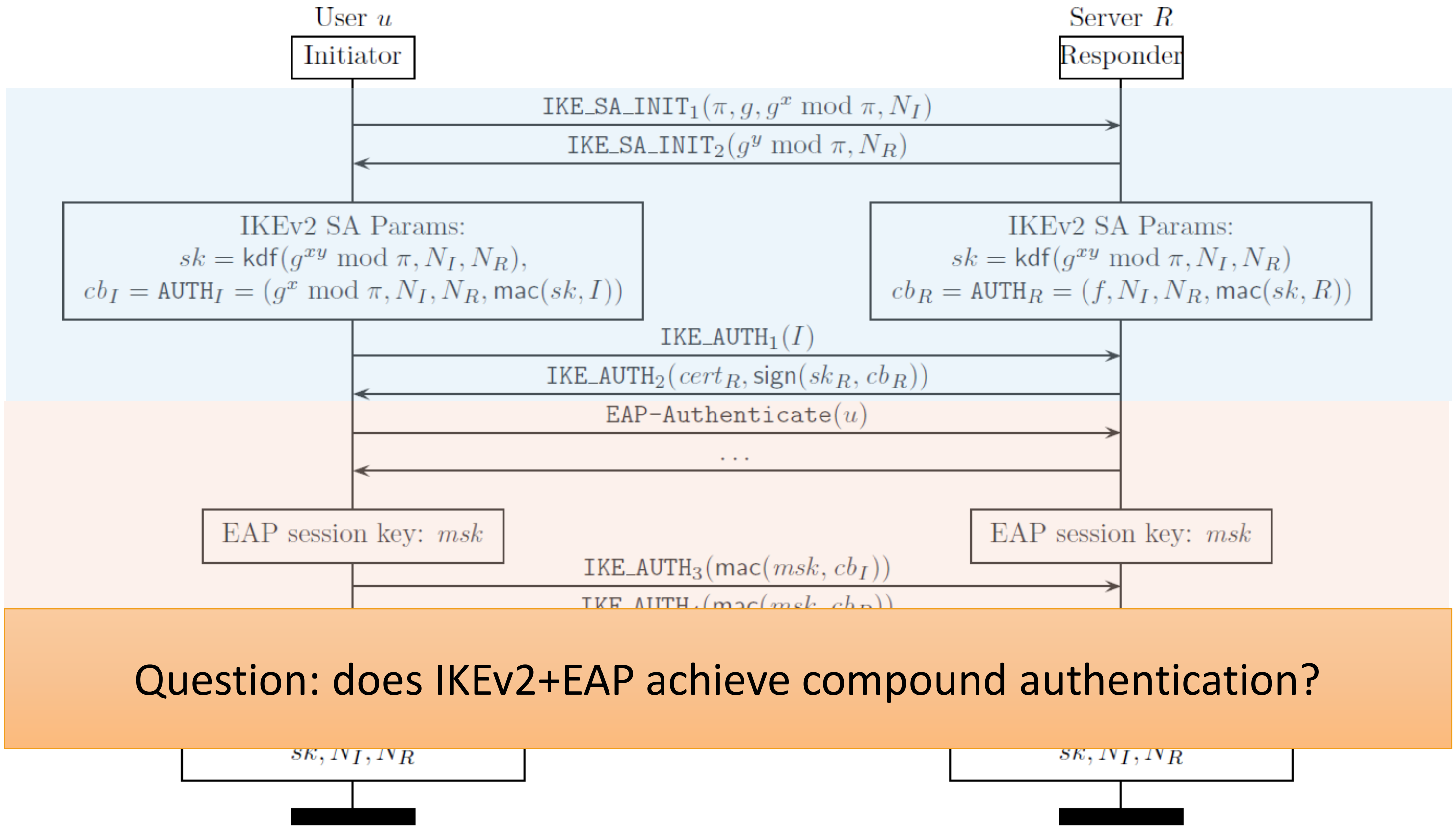
- Principal a completes the sequence of protocol instance $[l_1, \ldots, l_n]$
- Peer b sent a non-compromised credential in **some instance** l_i
- Session secrets in l_i have not been leaked

Then:

- b is not the attacker
- **For all j in [1,n]**, the dual instance l_j' ran by b agrees with l_j

NDSS '15 SYMPOSIUM

IPSec Example: IKEv2+EAP



Question: does IKEv2+EAP achieve compound authentication?

Small Subgroup Confinement Attacks

- Channel binding of IKEv2 based on Diffie-Hellman share and nonces:
  - $cb_I = (g^x mod\ \pi, n_i, n_r, MAC(g^{xy} mod\ \pi, I))$
  - (n_i, n_r) are nonces, (π, g) are Diffie-Hellman parameters
- What if the order g^x of is small in $< \pi >$?
  - Initiator can pick π, g, x such that g^x has a small order
- IKEv2 forbids 0, 1, -1 but allows other small subgroups
  - MitM can synchronize *cb* on both sides
  - **Fact: IKEv2+EAP doesn't achieve compound authentication**
- See paper for similar attacks on TLS-SRP and TLS-ECDHE on Curve25519

Small Subgroup Confinement Attacks

- If channel binding depends on public parameters + Diffie-Hellman shares, **improper DH validation breaks compound authentication**
  - "But these exclusions are unnecessary for Diffie-Hellman." – D. Berstein on the order 8 subgroup of Curve25519 allegedly not requiring validation

- If a peer can pick an arbitrary group (e.g. TLS-DHE) validation may be hard. Is it safer to use a transcript hash as channel binding?

Transcript Synchronization Attacks via Resumption

- Transcript hash may not authenticate **all session parameters** during resumption or re-keying
  - TLS: resumption only proves agreement on PMS; **can be synchronized** (3HS).
  - IKEv2: **resumption similar to TLS ticket resumption**; results in impersonation attack if IKEv2 re-authentication is supported (rare in practice).
  - *Using keys as credentials is dangerous!*

Formal Evaluation of Channel Bindings

- We create **ProVerif models of composed authentication schemes** and evaluate whether they **satisfy agreement and compound authentication**
- In addition to credential compromise, we model **small subgroup confinement** attacks by adding a constructor for bad elements that is invariant by exponentiation: $DHExp(badDH(gr), y) = badDH(gr)$.

Structure of Models and Queries

```
let initiator() =
 ... (* Model of initial key−exchange *)
 insert idb(l,ci,cr,params,sk)
| get idb(l,ci,cr,params,sk);
 ... (* Model of subsequent key−exchange *)
 insert idb(l',ci',cr',params',sk')
| ... (* Model of other subsequent key−exchange *)
```

```
process
  (* Responder credential generation *)
  new rsec:privkey; let rpub = pk(rsec) in out(net,rpub);
  (!initiator() | !responder(rpub,rsec))
```

Agreement Queries

```
query inj−event InitiatorEnd(pk(s),params,sk) =>
      inj−event ResponderBegin(pk(s),params,sk) || attacker(s)

query inj−event ResponderEnd(pk(s),params,sk) =>
      inj−event InitiatorBegin(pk(s),params,sk) || attacker(s).
```

Compound Authentication

```
query inj−event Compound_ResponderEnd(pk(s),p,sk,log) =>
  inj−event Compound_InitiatorBegin(pk(s),p,sk,log) || attacker(s).
```

Results

Model	Synchronization	Agreement (I)	Agreement (R)	Compound Auth	Verification Time
SSH + AUTH	None	✓ (after explicit key confirmation)	✓	✓	1.9 s
SSH + AUTH + Rekey	None	✓ (after explicit key confirmation)	✓	✗	1.9 s
SSH + AUTH + Rekey (cumulative hash)	None	✓ (no explicit key confirmation)	✓	✓	0.6 s
TLS with Ren./Res.	sid, ms, cr, sr	✓	N/A	N/A	1.3 s
TLS + SCRAM	sid, ms, cr, sr	✓	✓	✗	15.6 s
TLS + SCRAM (session hash)	None	✓	✓	✓	21.6 s

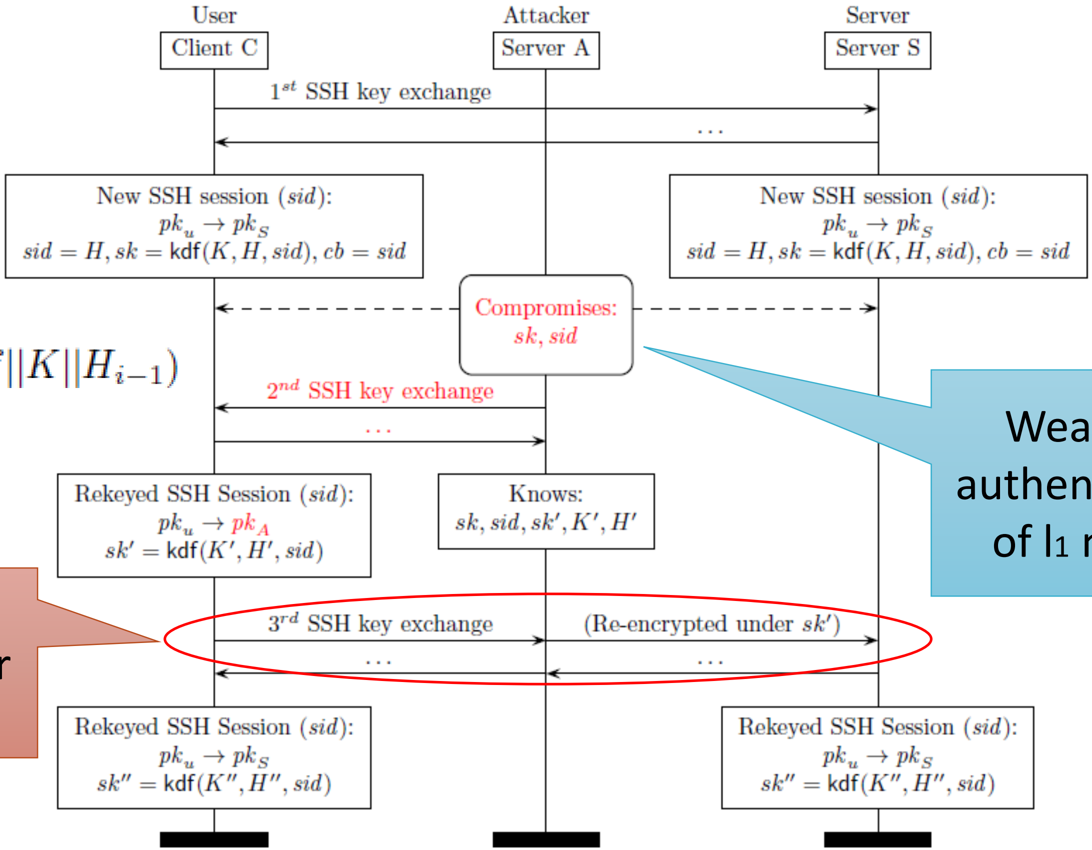
NDSS '15 SYMPOSIUM

http://prosecco.inria.fr/projects/channelbindings/

SSH Triple Exchange Attack

New proposal:

$$H_0 = \epsilon$$
$$H_i = hash(log||pk_S||e||f||K||H_{i-1})$$
$$sk_i = kdf^{SSH}(K, H_i)$$



Weak compound authentication: secret of l_1 must not leak

Host key from S is honest but 2nd peer was malicious