(DE-) CONSTRUCTING TLS 1.3

[paper published at indocrypt 2015]

Markulf Kohlweiss  
Ueli Maurer  
Cristina Onete  
**Björn Tackmann**  
Daniele Venturi  

Microsoft Research  
ETH Zürich  
INSA/IRISA Rennes  
UC San Diego  
Sapienza Univ. of Rome  

TRON Workshop @NDSS, San Diego, 21 February 2016
we prove the security of:

- signature-based Diffie-Hellman mode
- version 9 of the draft, October 2015
- basic security (honest server, no forward secrecy, no client auth, no downgrade analysis)
- caveat: we do not encrypt the certificate
- … but we learn more than just the facts during the analysis!
WHAT DOES TLS GIVE US?
WHAT DOES TLS GIVE US?

- bidirectional communication
- tls fragments of up to $2^{14}$ bytes
- attacker may learn message length
- attacker may interrupt the channel between fragments
WHAT DOES TLS ASSUME?

• bidirectional communication
• messages are tls fragments of up to $2^{14} + 256$ bytes
• attacker controls communication
• “functional” pki
PROTOCOL STACKS
PROTOCOL STACKS

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Internetwork</td>
</tr>
<tr>
<td>Media Access</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
THE DECONSTRUCTION

pki, network

server

signature

nonce

exchange

dh exchange

key derivation

authenticated encryption

secure channel
MODULARITY

Composable analysis of security mechanisms simplifies modular design of protocols.
SERVER SIGNATURE

signature

certificate

assumes:

\[ \text{signature} = \text{cert, sign}() \]
SERVER SIGNATURE

constructs:

server-authenticated message exchange

unauthenticated communication

is still available:
NONCE EXCHANGE

nonces

= signature

certificate

certificate

client nonce

server nonce
NONCE EXCHANGE

constructs:

nonces shared between client and server

still available:
DIFFIE-HELLMAN EXCHANGE

g^a

signature
certificate

dh

signature
certificate
DIFFIE-HELLMAN EXCHANGE

constructs:

server-authenticated premaster key

still available:
KEY COMPUTATIONS

local computation on

hkdf keys

hkdf fin

nonces
dh
signature
certificate

hkdf keys
KEY COMPUTATIONS

constructs:

server-authenticated traffic (etc.) keys

finished messages
PAYLOAD PROTECTION

data protected with
hkdf keys

data protected with
hkdf fin

ae payload

ae handshake

hkdf keys

hkdf fin
PAYLOAD PROTECTION

constructs:

server-authenticated secure channel

[badertscher, matt, maurer, rogaway, t., provsec 2015]
IS TLS 1.3 READY OR NOT?
WHAT ABOUT LEGO?
INSIGHTS FROM ANALYSIS

• we can make the proof work…

• separation of keys simplifies analysis

• “hashing everything” impedes modularization

• mutual dependencies (e.g., certificate/signature/dh exchange) do as well

• usefulness of nonces/finished messages unclear in our analysis