The Anatomy of Transport Layer Security

Joshua D. Guttman
Worcester Polytechnic Institute
The MITRE Corporation

March 2013
Bertinoro International Spring School

Thanks to the US National Science Foundation, under grant 1116557

guttman@wpi.edu

Transport-Layer Security

- Provide secure sessions between a client C and a server S
- Two main layers:
  - Record layer transports a sequence of pieces of data
  - Handshake layer agrees on keys to use in record layer
- Most communication happens in record layer
- Most of the interest is in the handshake protocol

The Record Layer

- Breaks stream of data into records
- For the i^{th} record t, uses a key mk for a Message Authentication Code

$$\text{HMAC}(mk, (i, t))$$

The Record Layer

- Breaks stream of data into records
- For the i^{th} record t, uses a key mk for a Message Authentication Code

$$\text{HMAC}(mk, (i, t))$$

- Integrity guarantee on contents t and position in stream i

The Record Layer

- Breaks stream of data into records
- For the i^{th} record t, uses a key mk for a Message Authentication Code

$$\text{HMAC}(mk, (i, t))$$

- Integrity guarantee on contents t and position in stream i
- Encrypts this and t using an ek

$$\{\!|\, t,\ \ \text{HMAC}(mk, (i, t))\, |\!\}_{ek}$$

The Record Layer

- Breaks stream of data into records
- For the i^{th} record t, uses a key mk for a Message Authentication Code

$$\text{HMAC}(mk, (i, t))$$

- Integrity guarantee on contents t and position in stream i
- Encrypts this and t using an ek

$$\{\!|\, t,\ \ \text{HMAC}(mk, (i, t))\, |\!\}_{ek}$$

- This provides confidentiality for whole unit

The Record Layer

- Breaks stream of data into records
- For the i^{th} record t, uses a key mk
  for a Message Authentication Code

$$\text{HMAC}(mk, (i, t))$$

- Integrity guarantee on contents t
  and position in stream i
- Encrypts this and t using an ek

$$\{\!|\, t,\ \ \text{HMAC}(mk, (i, t))\, |\!\}_{ek}$$

- This provides confidentiality for whole unit
- Requires 2 keys, mk and ek
  actually, two keys in each direction $C \to S$ and $S \to C$

Record Layer Security Goals

Confidentiality Adversary should learn nothing of contents of stream $C \rightarrow S$ or $S \rightarrow C$

Record Layer Security Goals

Confidentiality Adversary should learn nothing of contents of stream $C \rightarrow S$ or $S \rightarrow C$

Session Independence No principal $P \neq C, S$ should learn anything of contents of stream $C \rightarrow S$ or $S \rightarrow C$

Record Layer Security Goals

Confidentiality Adversary should learn nothing of contents of stream $C \to S$ or $S \to C$

Session Independence No principal $P \neq C, S$ should learn anything of contents of stream $C \to S$ or $S \to C$

Integrity The stream of data records received $\langle r_i \rangle_{i<\ell}$ should be an initial subsequence of those sent $\langle s_j \rangle_{j<\ell'}$

$$\ell \leq \ell' \quad \text{and} \quad r_k = s_k \quad \text{for all} \quad k < \ell$$

Record Layer Security Goals

Confidentiality Adversary should learn nothing of contents of stream $C \to S$ or $S \to C$

Session Independence No principal $P \neq C, S$ should learn anything of contents of stream $C \to S$ or $S \to C$

Integrity The stream of data records received $\langle r_i \rangle_{i<\ell}$ should be an initial subsequence of those sent $\langle s_j \rangle_{j<\ell'}$

$$\ell \leq \ell' \quad \text{and} \quad r_k = s_k \quad \text{for all} \quad k < \ell$$

Main inferred goals for Handshake layer:

Provide undisclosed keys mk, ck in each direction
Must be distinct in all sessions

The Handshake Protocol

Main Ideas (bilateral mode)

- C chooses the session secret *pms*
  the pre-master secret
- Confidentiality: encrypt *pms* with S's public encryption key
- S's authentication of C: C signs a msg
- Keys $\text{pubk}(S), \text{pubk}(C)$ are certified by a Certificate Authority
- Session property: Server creates a nonce r_s
  - Client also creates a nonce r_c
  - Allows *pms* reuse in some cases

  nonces contribute to keys

The Handshake

$C \to S$: r_c plus supplement

$S \to C$: r_s plus supplement

$S \to C$: $[\![\, \text{cert } S,\ \text{pubk}(S) \,]\!]_{\text{CA}}$ uses $\text{sk}(S)$

$C \to S$: $[\![\, \text{cert } C,\ \text{pubk}(C) \,]\!]_{\text{CA}}$

$C \to S$: $\{\!|\, \text{cl\_ver } \mathit{pms} \,|\!\}_{\text{pubk}(S)}$

$C \to S$: $[\![\, \text{Hash}(\mathit{previous\ msgs}) \,]\!]_{\text{sk}(C)}$

The Core Protocol

TLS subprotocol 0

$$C \rightarrow S: \ \{\!|\, \mathsf{cl\_ver}\ pms \,|\!\}_{\mathsf{pubk}(S)}$$

The Core Protocol

TLS subprotocol 0

$C \rightarrow S$: $\{\!|\, \mathsf{cl\_ver}\ pms \,|\!\}_{\mathsf{pubk}(S)}$

Ensures to C that pms undisclosed
assuming $\mathsf{sk}(S)$ uncompromised

TLS subprotocol 1

$S \to C$: $[\![\, \mathsf{cert}\ S,\ \mathsf{pubk}(S) \,]\!]_{\mathsf{CA}}$

$C \to S$: $\{\!|\, \mathsf{cl\_ver}\ \mathit{pms} \,|\!\}_{\mathsf{pubk}(S)}$

TLS subprotocol 1

$$CA \to C: \ [\![\text{cert } S,\ \text{pubk}(S)]\!]_{CA}$$

$$C \to S: \ \{\!| \text{cl\_ver } pms |\!\}_{\text{pubk}(S)}$$

TLS subprotocol 2

$$\mathsf{CA} \to C\colon\ [\![\, \mathsf{cert}\ S,\ \mathsf{pubk}(S)\,]\!]_{\mathsf{CA}}$$

$$\mathsf{CA} \to S\colon\ [\![\, \mathsf{cert}\ C,\ \mathsf{pubk}(C)\,]\!]_{\mathsf{CA}}$$

$$C \to S\colon\ \{\!|\, \mathsf{cl\_ver}\ \mathit{pms}\,|\!\}_{\mathsf{pubk}(S)}$$

$$C \to S\colon\ [\![\, \mathsf{Hash}(\mathit{previous}\ \mathit{msgs})\,]\!]_{\mathsf{sk}(C)}$$

TLS subprotocol 3

$$\begin{aligned}
CA \to C&: \ [\![\, \mathsf{cert}\ S,\ \mathsf{pubk}(S)\,]\!]_{CA} \\
CA \to S&: \ [\![\, \mathsf{cert}\ C,\ \mathsf{pubk}(C)\,]\!]_{CA} \\
C \to S&: \ \{\!|\, \mathsf{cl_ver}\ \mathit{pms}\,|\!\}_{\mathsf{pubk}(S)} \\
C \to S&: \ [\![\, \mathsf{Hash}(\mathit{previous}\ \mathit{msgs})\,]\!]_{\mathsf{sk}(C)}
\end{aligned}$$

TLS subprotocol 3

$$\mathrm{CA} \to C\colon\ [\![\, \mathrm{cert}\ S,\ \mathrm{pubk}(S)\,]\!]_{\mathrm{CA}}$$

$$\mathrm{CA} \to S\colon\ [\![\, \mathrm{cert}\ C,\ \mathrm{pubk}(C)\,]\!]_{\mathrm{CA}}$$

$$C \to S\colon\ \{\!|\, \mathrm{cl\_ver}\ \mathit{pms}\,|\!\}_{\mathrm{pubk}(S)}$$

$$C \to S\colon\ [\![\, \mathrm{Hash}(\mathit{previous}\ \mathit{msgs})\,]\!]_{\mathrm{sk}(C)}$$

Same as subprotocol 2

TLS subprotocol 3

$CA \rightarrow C$: $[\![\text{cert } S, \text{ pubk}(S)]\!]_{CA}$

$CA \rightarrow S$: $[\![\text{cert } C, \text{ pubk}(C)]\!]_{CA}$

$C \rightarrow S$: $\{\!| \text{cl\_ver } pms |\!\}_{\text{pubk}(S)}$

$C \rightarrow S$: $[\![\text{Hash}(\textit{previous msgs})]\!]_{\text{sk}(C)}$

```
(defrole certificate_auth
  (vars (subject_name ca name))
    (trace
      (send (cert subject_name (pubk subject_name) (privk ca))
  (non-orig (privk subject_name)))
```

The Handshake

$C \to S$: r_c	plus supplement
$S \to C$: r_s	plus supplement
$S \to C$: $[\![\, \text{cert } S,\ \text{pubk}(S) \,]\!]_{\text{CA}}$	uses sk(S)
$C \to S$: $[\![\, \text{cert } C,\ \text{pubk}(C) \,]\!]_{\text{CA}}$	
$C \to S$: $\{\!\|\, \text{cl\_ver } \mathit{pms} \,\|\!\}_{\text{pubk}(S)}$	
$C \to S$: $[\![\, \text{Hash}(\mathit{previous\ msgs}) \,]\!]_{\text{sk}(C)}$	

The TLS Key Derivations

$$ms = \text{Hash}(pms,\ \text{PreMasterSec}\ ,\ r_c,\ r_s)$$

The TLS Key Derivations

$$ms = \text{Hash}(pms,\ \text{PreMasterSec}\ ,\ r_c,\ r_s)$$

$$cm = \text{Hash}(\text{ClientMAC}\ ms)$$

The TLS Key Derivations

$$ms = \text{Hash}(pms,\ \text{PreMasterSec}\ ,\ r_c,\ r_s)$$

$$cm = \text{Hash}(\text{ClientMAC}\ ms)$$

$$ce = \text{Hash}(\text{ClientEnc}\ ms)$$

The TLS Key Derivations

$$ms = \text{Hash}(pms,\ \text{PreMasterSec}\ ,\ r_c,\ r_s)$$

$$cm = \text{Hash}(\text{ClientMAC}\ ms)$$

$$ce = \text{Hash}(\text{ClientEnc}\ ms)$$

$$sm = \text{Hash}(\text{ServerMAC}\ ms)$$

$$se = \text{Hash}(\text{ServerEnc}\ ms)$$

Record Layer Security Goals

Confidentiality Adversary should learn nothing of contents of stream $C \to S$ or $S \to C$

Session Independence No principal $P \neq C, S$ should learn anything of contents of stream $C \to S$ or $S \to C$

Integrity The stream of data records received $\langle r_i \rangle_{i<\ell}$ should be an initial subsequence of those sent $\langle s_j \rangle_{j<\ell'}$

$$\ell \leq \ell' \quad \text{and} \quad r_k = s_k \quad \text{for all} \quad k < \ell$$

Main inferred goals for Handshake layer:

Provide undisclosed keys mk, ck in each direction
Must be distinct in all sessions