

Is TLS secure?

P. Morrissey N.P. Smart B. Warinschi

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB
United Kingdom.

January 31, 2008

Outline

Introduction

Pre-Master Secret Security Model

Master Secret Security Model

Application Key Security Model

Introduction

Introduction to TLS

SSL/TLS is probably the most deployed/used security protocol on the Internet.

Designed a relatively long time ago.

- ▶ No “cryptographic” proof of security
- ▶ Some proofs in the Dolev-Yao/Symbolic model

Question: What cryptographic security properties does it have ?

Question: Can we shed some light on its design ?

SSL/TLS Stack

SSL/TLS first agrees a pre-master secret key

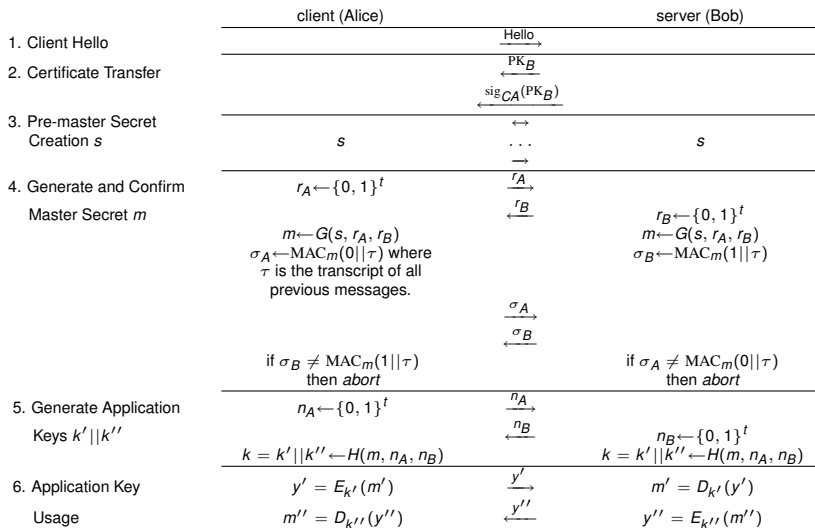
Then a master secret is derived from this using shared randomness

- ▶ This key is confirmed using a Mac of the transcript under this key

Application keys are then derived from the master secret key via more shared randomness

Application keys are not confirmed, but are then used in the protocol

SSL/TLS Stack



Security Models

The most successful security model for key agreement protocols has been one based on the work of Bellare and Rogaway.

A key is considered secure if it cannot be distinguished from a random key under attacks which allow parties to be adaptively corrupted and in which the adversary controls the network.

All used PMS sub-protocol do not meet this requirement.

The master secret key protocol does not meet this requirement

- ▶ Key cannot be indistinguishable as it is used in the Mac

So what security gaurantee's do we obtain?

Pre-Master Secret Security Model

PMS Security Model

Parties modelled as a series of oracles \mathcal{O} .

Adversary can execute a series of commands:

- ▶ $\text{NewSession}(U, \text{role})$: Creates a new oracle \mathcal{O} for user U playing the role $\text{role} \in \{\text{initiator}, \text{responder}\}$.
- ▶ $\text{Send}(\mathcal{O}, \text{msg})$: Sends a message to \mathcal{O} and obtains the response
- ▶ $\text{Corrupt}(U)$: Obtains the long term key of U , all oracles associated with U have power transferred to the adversary.
- ▶ $\text{Check}(\mathcal{O}, s)$: For an oracle which has accepted a PMS, this checks whether s is the pre-master secret obtained.

The main thing here which is different from Bellare-Rogaway is the use of the Check command rather than a Reveal command.

PMS Security Model

An oracle is said to be fresh if

- ▶ It has accepted a pre-master secret
- ▶ It thinks it shares a key with V and V is not corrupted.
- ▶ It is not corrupted itself.

Fresh oracles define the oracles which we do not want the adversary to be able to attack.

He should not be able to break keys of uncorrupted players essentially.

PMS Security Model

We define security for PMS via a one-way as opposed to an IND game:

Security defined by a game $\text{Exec}_{\mathcal{A}, \Pi}^{\text{OW-PMS}}(t)$ between an adversary \mathcal{A} and a challenger \mathcal{C} for the protocol Π .

1. \mathcal{C} , generates public/secret key pairs for each user U , and returns the public keys to \mathcal{A} .
2. \mathcal{A} is allowed to make as many NewSession, Send, Check, and Corrupt queries as it likes.
3. At some point \mathcal{A} outputs a pair (\mathcal{O}^*, s^*) , where \mathcal{O}^* is one of \mathcal{A} 's oracles, and $s^* \in \mathcal{S}_{\text{PMS}}$.

PMS Security Model

Adversary \mathcal{A} wins the $\text{Exec}_{\Pi, \mathcal{A}}^{\text{OW-PMS}}$ game if

- ▶ \mathcal{O}^* is fresh.
- ▶ $s^* = s_{\mathcal{O}^*}$.

In other words the adversary determines the pre-master secret key held by \mathcal{O}^* .

- ▶ This is security in a one-way sense

We write

$$\text{Adv}_{\mathcal{A}, \Pi}^{\text{OW-PMS}}(t) = \Pr[\mathcal{A} \text{ wins}],$$

for the advantage of \mathcal{A} in winning the $\text{Exec}_{\mathcal{A}, \Pi}^{\text{OW-PMS}}$ game.

Deterministic Key Transport

Theorem

If Enc is a OW-CPA secure deterministic encryption scheme, then the pre-master secret key agreement protocol $\text{PMK}(\text{Enc})$ is secure.

We show that for any adversary A against the $\text{PMK}(\text{Enc})$ there exists an adversary B against OW-CPA security of Enc such that:

$$\text{Adv}_{\mathcal{A}, \text{PMK}(\text{Enc})}^{\text{OW-PMS}}(t) \leq (n_P \cdot n_S \cdot (n_P + n'_P)) \cdot \text{Adv}_{\mathcal{B}, \text{Enc}}^{\text{OW-CPA}}(t).$$

This theorem means that plain textbook RSA can be used in the context of SSL/TLS.

- ▶ However, in practice SSL/TLS uses a randomized version of RSA...

Probabilistic Key Transport

Theorem

If Enc is a OW-CCA secure randomized encryption scheme, then the pre-master secret key agreement protocol PMK(Enc) is secure.

We show that for any adversary \mathcal{A} against PMK(Enc), there exists an adversary \mathcal{B} against the OW-CCA security of Enc such that

$$\text{Adv}_{\mathcal{A}, \text{PMK}(\text{Enc})}^{\text{OW-PMS}}(t) \leq (n_P \cdot n_S \cdot (n_P + n'_P)) \cdot \text{Adv}_{\mathcal{B}, \text{Enc}}^{\text{OW-CCA}}(t).$$

Signed Diffie–Hellman

Theorem

Let \mathbb{G} be cyclic group for which the gap-Diffie-Hellman assumption holds and let Sig be a secure public key signature scheme. Then $\text{PMK}(\text{Sig}, \mathbb{G})$ is a secure pre-master key agreement protocol.

We show that for any adversary \mathcal{A} against $\text{PMK}(\text{Sig}, \mathbb{G})$ there exists an algorithm \mathcal{B} for the gap-Diffie–Hellman problem in \mathbb{G} and an adversary \mathcal{C} against Sig such that:

$$\text{Adv}_{\mathcal{A}, \text{PMK}(\text{Sig}, \mathbb{G})}^{\text{OW-PMS}}(t) < \text{Adv}_{\mathcal{B}, \mathbb{G}}^{\text{gap-DH}}(t) + n_P \cdot \text{Adv}_{\mathcal{C}, \text{Sig}}^{\text{SEF-CMA}}(t).$$

Commentary

Why are these theorems true:

- ▶ A OW-CPA encryption scheme is one-way, hence so is the PMS protocol.
- ▶ A OW-CCA encryption scheme is one-way, hence so is the PMS protocol.
- ▶ For a deterministic OW-CPA encryption scheme you can answer the Check queries as the scheme is deterministic.
- ▶ For a OW-CCA encryption scheme you can answer the Check queries via the decryption oracle.

- ▶ For signed Diffie–Hellman the Check queries are answered by the DDH oracle in the Gap-DH assumption.
- ▶ The PMS scheme is one-way since either you break the Gap-DH assumption or you manage to forge a signature.

The PMS protocols are secure against man-in-the-middle attacks, but are not secure against unknown-key-share attacks.

Master Secret Security Model

MS Security Model

The master key agreement protocol derived from a pre-master key agreement protocol protects against unknown-key-share attacks, and provides explicit key confirmation.

Security defined as before by a game, two changes:

- ▶ $\text{Check}(\mathcal{O}, m)$: This now checks whether m is the master secret, not the pre-master secret belonging to \mathcal{O} .
- ▶ $\text{Reveal}(\mathcal{O})$: This is a new query which allows the adversary to obtain the master secret belonging to an oracle.

Freshness is now augmented to also mean that no revealed oracle can have had a matching conversation with \mathcal{O} .

- ▶ Matching conversation is a standard notion due to Bellare and Rogaway.
- ▶ Means two oracles have essentially the same incoming and outgoing message flows.

MS Security Model

Security still defined by a one-way sense

- ▶ Has to be due to the Mac.

Let

- ▶ Π denote a pre-master key agreement protocol
- ▶ G the hash function
- ▶ Mac a Mac function

then $(\Pi; \text{MKD}_{\text{SSL}}(\text{Mac}, G))$ is the derived master key agreement protocol.

MS Security Model

Theorem

Let Π be a secure pre-master agreement protocol, Mac be a secure message authentication code, and G a random oracle. Then $(\Pi; \text{MKD}_{\text{SSL}}(\text{Mac}, G))$ is a secure master-key agreement protocol.

In particular:

$$\text{Adv}_{\mathcal{A}, (\Pi; \text{MKD}_{\text{SSL}}(\text{Mac}, G))}^{\text{OW-MS}}(t) < \text{Adv}_{\mathcal{B}, \Pi}^{\text{OW-PMS}}(t) + (n_P \cdot n_S \cdot (n_P + n'_P)) \cdot \text{Adv}_{\mathcal{C}, \text{Mac}}^{\text{OW-CMA}}(t).$$

Note breaking the Mac here means recovering the key used to create a specific Mac on a specific message.

Commentary

Security against unknown-key-share attacks provided due to the Reveal queries.

Reveal queries can be answered due to the Check queries of the underlying PMS protocol and the random oracle G .

Can also show that the master secret protocol satisfies the standard definition for key confirmation.

- ▶ For details see our paper
- ▶ Here we require the standard security model of a Mac

Application Key Security Model

Application Keys

A number of application keys can be derived from a master secret key

- ▶ Allows rekeying without expensive public key operations.

Application keys are not explicitly confirmed, only confirmed implicitly in any resulting application.

Application keys can (essentially) be combined with any application.

- ▶ Application keys are secure in the standard IND sense, as opposed to the previous one-way sense.

Application Keys

In the resulting application need to allow for adaptive corruptions of symmetric encryption keys

- ▶ Needs a small alteration to the standard IND-CCA model for symmetric encryption in the multi-user setting.

In our paper we consider all the previous issues and show that if one takes a secure master-key agreement protocol and derives the application keys as is done in SSL/TLS then the resulting final protocol is secure.

Application Keys

However, the devil is in how it is all implemented....

- ▶ Our results are in the ROM.
- ▶ Assume proper IND-CCA secure symmetric encryption is used as the application.
- ▶ Error messages do not leak information (Bleichenbacher's attack)
- ▶ All subcomponents, e.g. Mac's, RSA, DH, etc are secure.

Main results though:

1. One can provide a cryptographic proof of a large protocol like SSL/TLS by breaking it into smaller component and using game-hopping techniques.
2. Sheds some light on the design decisions of SSL/TLS, e.g. explains what properties each stage of the protocol provide.

Thank You