Analysis of the TLS 1.0 handshake protocol with AVISPA

Kaspar-David Buss and Moritz Krebbel

January 26, 2016

Overview

1. Introduction
2. The TLS Handshake Protocol
   - Intruder model and security goals
   - AVISPA model
3. Insecure variants
4. Runtime statistics
   - Results for specific tools
   - Comparison
5. Conclusion

Motivation

Transport Layer Security (TLS) provides

- confidentiality
- authentication

between two agents

Motivation

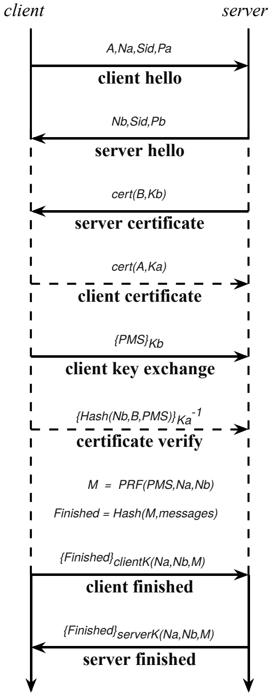
Transport Layer Security (TLS) provides

- confidentiality
- authentication

between two agents

TLS Handshake Protocol serves to

- negotiate encryption algorithms
- negotiate secret symmetric keys
- authenticate agents to each other



1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : \{B, K_b\}_{inv(K_s)}$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}_{K_b}$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$
7: $A \rightarrow B : \{Finished\}_{Keygen(A,N_a,N_b,M)}$
8: $B \rightarrow A : \{Finished\}_{Keygen(B,N_a,N_b,M)}$

Intruder model and security goals

Intruder model: Dolev-Yao
Intruder knowledge: public keys, own private key, own certificate, functions

Goals

- secrecy of symmetric client key
- secrecy of symmetric server key
- Alice authenticates Bob
- Bob authenticates Alice

AVISPA model

Simplified the protocol as follows

- fixed encryption preferences $P_b = P_a$
- only client-authenticated handshake
- master secret and key calculation using hash function
- no restart and renegotiation of session
- no *Hello Request* from server

One symmetric key

1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : \{B, K_b\}_{inv(K_s)}$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}K_b$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$
7: $A \rightarrow B : \{Finished\}_{Keygen(N_a,N_b,M)}$
8: $B \rightarrow A : \{Finished\}_{Keygen(N_a,N_b,M)}$

One symmetric key

1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : \{B, K_b\}_{inv(K_s)}$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}K_b$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$
7: $A \rightarrow B : \{Finished\}_{Keygen(N_a,N_b,M)}$
8: $B \rightarrow A : \{Finished\}_{Keygen(N_a,N_b,M)}$

1: $A \rightarrow I : A, N_a, Sid, P_a$
2: $I \rightarrow B : A, N_i, Sid, P_i$
3: $B \rightarrow I : N_b, Sid, P_i$
4: $B \rightarrow I : \{B, K_b\}_{inv(K_s)}$
5: $I \rightarrow A : N_i, Sid, P_a$
6: $I \rightarrow A : \{B, K_b\}_{inv(K_s)}$
7: $A \rightarrow I : \{A, K_a\}_{inv(K_s)}$
8: $A \rightarrow I : \{PMS\}_{K_b}$
9: $A \rightarrow I : H(N_i, B, PMS)_{inv(K_a)}$
10: $A \rightarrow I : Finished_{Keygen(N_a,N_b,M)}$
11: $I \rightarrow A : Finished_{Keygen(N_a,N_b,M)}$

Without certificate check

1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : B, K_b$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}K_b$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$
7: $A \rightarrow B : \{Finished\}_{Keygen(A,N_a,N_b,M)}$
8: $B \rightarrow A : \{Finished\}_{Keygen(B,N_a,N_b,M)}$

Without certificate check

1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : B, K_b$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}K_b$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$
7: $A \rightarrow B : \{Finished\}_{Keygen(A,N_a,N_b,M)}$
8: $B \rightarrow A : \{Finished\}_{Keygen(B,N_a,N_b,M)}$

1: $A \rightarrow I : A, N_a, Sid, P_a$
2: $I \rightarrow A : N_i, Sid, P_i$
3: $I \rightarrow A : B, K_i$
4: $A \rightarrow I : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow I : \{PMS\}_{K_i}$
6: $A \rightarrow I : H(N_i, B, PMS)_{inv(K_a)}$
7: $A \rightarrow I : Finished_{Keygen(A,N_a,N_b,M)}$
8: $I \rightarrow A : Finished_{Keygen(B,N_a,N_b,M)}$

Without finish messages

1: $A \rightarrow B : A, N_a, Sid, P_a$
2: $B \rightarrow A : N_b, Sid, P_b$
3: $B \rightarrow A : \{B, K_b\}_{inv(K_s)}$
4: $A \rightarrow B : \{A, K_a\}_{inv(K_s)}$
5: $A \rightarrow B : \{PMS\}_{K_b}$
6: $A \rightarrow B : H(N_b, B, PMS)_{inv(K_a)}$

Without finish messages

1: $A \to B : A, N_a, Sid, P_a$
2: $B \to A : N_b, Sid, P_b$
3: $B \to A : \{B, K_b\}_{inv(K_s)}$
4: $A \to B : \{A, K_a\}_{inv(K_s)}$
5: $A \to B : \{PMS\}_{K_b}$
6: $A \to B : H(N_b, B, PMS)_{inv(K_a)}$

1: $A \to I : A, N_a, Sid, P_a$
2: $I \to B : A, N_i, Sid, P_i$
3: $B \to I : N_b, Sid, P_i$
4: $B \to I : \{B, K_b\}_{inv(K_s)}$
5: $I \to A : N_i, Sid, P_a$
6: $I \to A : \{B, K_b\}_{inv(K_s)}$
7: $A \to I : \{A, K_a\}_{inv(K_s)}$
8: $A \to I : \{PMS\}_{K_b}$
9: $A \to I : H(N_i, B, PMS)_{inv(K_a)}$

OFMC

	result	parse time	search time	visited nodes	depth
standard	safe	<0.01	0.43	332	10
one sym key	unsafe	<0.01	0.02	19	3
wo cert check	unsafe	<0.01	0.01	7	2
wo finish	unsafe	<0.01	0.01	6	2

CL-AtSe

Depth-first search is standard; results for breadth-first search in brackets

	result	analysed	reachable	translation time	computation time
standard	safe	17192	11058	<0.01	0.62
one sym key	unsafe	15(78)	13(68)	<0.01	<0.01
wo cert check	unsafe	11(66)	11(60)	<0.01	<0.01
wo finish	unsafe	11(23)	9(21)	<0.01	<0.01

SATMC

Always returns `SAFE`
Compilation time always about 3 seconds
Probably broken because

- *atomsNumber* is 0
- *clausesNumber* is 0
- *stepsNumber* is 1
- *upperBoundReached* is *true*

TA4SP

Always returns `INCONCLUSIVE`
Regardless of settings, constant issues with *left-linearity*
Even if attack found $\Rightarrow$ No trace provided

Results extremely useless

Comparison

OFMC

- very reliable
- very fast

CL-AtSe

- just as reliable
- slightly slower

SATMC

- worse than useless
- extremely slow

TA4SP

- pretty quick...
- to determine that it doesn't work

Conclusion

In this project, we have

- modeled the TLS handshake protocol in AVISPA
- formulated security goals
- tested our model with regards to goals
- modeled variations on the protocol

Further possibilities

- model restart and renegotiation of session
- model *Hello Request* from server

Thank you for your attention

Thank you for your attention

Questions?