Attacking Multicast Group Key Management Protocols

Graham Steel and Alan Bundy

Multicast Key Management Protocols

Aim: To maintain a secure key for multicast within a group as agents join and leave

- Analysis of these protocols is challenging:

  Modelling the protocols, posing security conjectures, searching in the model created

- Aims of this talk:

  Demonstrate efficacy of CORAL approach

  Describe what modifications other tools would need to tackle these protocols

CORAL

Refutes incorrect inductive conjectures

Uses a method borrowing theory from ‘Proof by Consistency’
- a refutation complete method for proving inductive theorems

First-order version of Paulson model

By refuting a security property $\forall trace.P(trace)$, we obtain the attack as the instantiation of $trace$

Tested on several known attacks (from Clark-Jacob corpus)

New attacks on Asokan–Ginzboorg

Example - Tagdhiri Jackson

Originally proposed by Tanaka + Sato. T+J found flaws using Alloy + SAT checker, proposed improved protocol.

Flaw due to retention of old keys

However, their model did not include an active attacker!

CORAL used to model + attack the improved version

Tanaka-Sato/Taghdiri-Jackson

Join:

1. $M_i \rightarrow S : \{\!|\ \text{join}\ |\!\}_{K_{M_i}}$
2. $S \rightarrow M_i : \{\!|\ Ik_{M_i}, Gk(n) |\!\}_{K_{M_i}}$

Send:

1. $M_i \rightarrow S : \{\!|\ send(n) |\!\}_{Ik_{M_i}}$
2. $S \rightarrow M_i : \{\!|\ n', Gk(n') |\!\}_{Ik_{M_i}}$

Leave:

1. $M_i \rightarrow S : \{\!|\text{leave}|\!\}_{Ik_{M_i}}$
2. $S \rightarrow M_i : \{\!|\text{ack.leave}|\!\}_{Ik_{M_i}}$

   (and generate new key)

Receive:

1. $M_j \rightarrow S : \{\!|\ read(n) |\!\}_{Ik_{M_j}}$
2. $S \rightarrow M_j : \{\!|\ Gk(n') |\!\}_{Ik_{M_j}}$

Modelling the Protocol

- Want to keep model general wrt no. of agents, scenario...

  CORAL's inductive model ideal for this

- Importance of knowing who is in the group at all times

  Stored in trace

- Lots of fresh material needed

  Use of counter, heuristic

Security Properties

Pereira–Quisquater properties unsuitable

Need *multicast group authenticity*

Throughout the evolution of the group, non-members should not be accepted as group members – whether sending or receiving

Must make concrete conjectures in terms of trace

Difficult without allowing ‘transient security breach’ to count as an attack

Example

m(cons(sent(Mj,all,encr(hello(Y),Gk),Xgroup),

cons(sent(X,Mj,encr(pair(Gk,send(Sq2)),Ikey),Xgroup),

cons(sent(Mj,server,encr(send(Sq2),Ikey),Xgroup),

Trace))),Group,Keyseq,Tick)=true $\wedge$

eqagent(Mj,spy)=false $\wedge$

in(Gk,analz(Trace)=true $\wedge$

ingroup(triple(principal(spy),X3,X2),Xgroup,Newgp)=false

$\longrightarrow$

School of informatics

Attack on Taghdiri Jackson

5.	spy	$\rightarrow$	server	:	$\{\!\| send(1) \|\!\}_{ik(spy)}$
6.	server	$\rightarrow$	spy	:	$\{\!\| Gk(2), send(1) \|\!\}_{ik(spy)}$
7.	a	$\rightarrow$	server	:	$\{\!\| send(2) \|\!\}_{ik(a)}$
8.	server	$\rightarrow$	a	:	$\{\!\| Gk(2), send(2) \|\!\}_{ik(a)}$
9.	a	$\rightarrow$	all	:	$\{\!\| hello(9) \|\!\}_{Gk(2)}$
10.	spy	$\rightarrow$	server	:	$\{\!\| leave \|\!\}_{ik(spy)}$
11.	server	$\rightarrow$	spy	:	$\{\!\| ackleave \|\!\}_{ik(spy)}$
12.	a	$\rightarrow$	server	:	$\{\!\| send(2) \|\!\}_{ik(a)}$
13.	spy	$\rightarrow$	a	:	$\{\!\| Gk(2), send(2) \|\!\}_{ik(a)}$
14.	a	$\rightarrow$	all	:	$\{\!\| hello(14) \|\!\}_{Gk(2)}$

Iolus

School of informatics

Join:

1. $M_i \rightarrow S : \{\!|\ \text{join}\ |\!\}_{K_{M_i}}$
2. $S \rightarrow M_i : \{\!|\ Ik_{M_i}, Gk(n) |\!\}_{K_{M_i}}$
3. $S \rightarrow \text{ALL} : \{\!|\ Gk_{n'} |\!\}_{Gk_n}$

Send:

1. $M_i \rightarrow \text{ALL} : \{\!|\ message |\!\}_{Gk(n)}$

Leave:

1. $M_i \rightarrow S : \{\!|\text{leave}|\!\}_{Ik_{M_i}}$
2. $S \rightarrow ALL : [\ \{\!|\ Gk_{n'} |\!\}_{Ik_{M_j}}\ \ldots] \ \forall j \neq i, M_j \in \text{group}$

School of informatics

Modelling Iolus

- For a general model, need lists for key update
  - Needed this before for Asokan–Ginzboorg
  - Straightforward in CORAL
- Control conditions become non-trivial
  - Must work out what the key update message is
  - Use recursive auxiliary function (as for A-G)
- No separate send/receive protocols
  - Makes posing conjectures easier

Attack on Iolus

9.	server	$\rightarrow$	s(a)	:	$\{\!\vert\, ik(11), Gk(11) \vert\!\}_{longtermK(s(a))}$
10.	a	$\rightarrow$	server	:	$\{\!\vert\, leave \vert\!\}_{ik(2)}$
11.	server	$\rightarrow$	all	:	$[\{\!\vert\, Gk(14) \vert\!\}_{ik(11)}, \{\!\vert\, Gk(14) \vert\!\}_{ik(5)}]$
12.	spy	$\rightarrow$	server	:	$\{\!\vert\, leave \vert\!\}_{ik(5)}$
13.	server	$\rightarrow$	all	:	$[\{\!\vert\, Gk(26) \vert\!\}_{ik(11)}]$
14.	spy	$\rightarrow$	all	:	$[\{\!\vert\, Gk(14) \vert\!\}_{ik(11)}, \{\!\vert\, Gk(14) \vert\!\}_{ik(5)}]$

Summary

Strengths

- Natural, general model in inductive formalism
- Could pose novel security properties
- Found 3 new attacks on 2 protocols

Weaknesses

- Slow - up to 3 hours
- Posing conjectures tricky

  though easier second time, and not just CORAL

What Was Required

- Arbitrary number of agents
- Lists
- Auxiliary functions
- Conjectures involving temporal properties

Further Work

- More group protocols, with Diffie-Hellman operations
- API attacks - Bond–Clulow

More details

http://homepages.inf.ed.ac.uk/s9808756/coral