On the Security of APKES

Rob Dallara and Rob Lewis

Department of Computer Science
University of North Carolina at Chapel Hill

Dec 2, 2014

Background

The emerging internet of things

- More embedded devices are being directly connected to the internet
- Huge opportunities for the future, but wide range of new vulnerable applications
- In particular, vulnerabilities may arise from protocols that embedded operating systems (such as Contiki) use.

6LoWPAN Basics

- 6LoWPAN is a protocol stack for integrating decentralized wireless networks of sensors with IPv6.
- Goal is to transmit packets between different embedded nodes securely and with energy efficiency.

Problems with Other Key Schemes

- Network-wide shared keys fail to provide security after the compromise of a single embedded node
- Fully pairwise key schemes take up too much memory on embedded nodes
- The use of public keys is too time and energy-consuming.

APKES Protocol

- Generation of shared secrets between two nodes
- Pairwise key establishment

$$
\begin{aligned}
& u : \text{Generate } R_u \text{ randomly} \\
u \rightarrow * & : \texttt{HELLO} \ \langle R_u \rangle \\
& v : \text{Generate } R_v \text{ randomly and wait for } T_w \leq M_w \\
& v : K_{v,u} = \text{see Table 1} \\
v \rightarrow u & : \texttt{HELLOACK} \ \langle R_u, R_v \rangle_{K_{v,u}} \\
& v : K'_{v,u} = \text{AES}(K_{v,u}, R_u \| R_v) \\
& u : K_{u,v} = \text{see Table 1} \\
& u : K'_{u,v} = \text{AES}(K_{u,v}, R_u \| R_v) \\
u \rightarrow v & : \texttt{ACK} \ \langle \rangle_{K'_{u,v}}
\end{aligned}
$$

Scyther

- Model protocols using Secure Protocol Description Language (SPDL)
- Roles not actors
- Characterization
- Pattern refinement
- Guaranteed termination
    - (Un/)bounded verification
    - Falsification w/ counter-example(s)

Verification Scope

- APKES only (No pluggable schemes)
  - using Scyther secret key infrastructure
- Did not model replay attacks
  - … or the frame counters used to mitigate them
- Security properties only
- Perfect cryptography
- Dolev-Yao attacker model

Verification Claims

- Secrecy of pairwise key
- Agreement
- Synchronization

Demo & Results

Claim				Status		Comments
APKES	U	APKES,i1	Secret {Ru,Rv}k(V,U)	Ok	Verified	No attacks.
		APKES,i2	Niagree	Ok	Verified	No attacks.
		APKES,i3	Nisynch	Ok	Verified	No attacks.
	V	APKES,V1	Secret {Ru,Rv}k(V,U)	Ok	Verified	No attacks.
		APKES,V2	Niagree	Ok	Verified	No attacks.
		APKES,V3	Nisynch	Ok	Verified	No attacks.

Done.

Future Work

- Pluggable scheme implemented for Contiki (LEAP)
  - APKES + LEAP fit w/ Scyther multi-protocol capability
- Replay attacks & mitigation measures
- Newer model checker (Tamarin)

Conclusions

- No attacks found on our APKES model
- However, other types of attacks may exist
  - Didn’t model replay attacks, or frame counters used to mitigate them
  - Failures of implementation
- Scyther is powerful & performant, but it’s important to understand its view of the world

Resources

- Our code - github.com/superdude264/Contiki_CoreSec_Verification
- 6LoWPAN Security: Adding Compromise Resilience to the 802.15.4 Security Sublayer (K. Krentz)
- APKES in Contiki - github.com/kkrentz/contiki/wiki
- Unbounded Verification, Falsification, and Characterization of Security Protocols by Pattern Refinement (C. Cremers)
- Scyther - cs.ox.ac.uk/people/cas.cremers/scyther

Thank You!

Q & A