Practical failure modes in protocol analysis

David Wagner

U.C. Berkeley

`daw@cs.berkeley.edu`

Introduction

- Abstract protocol notation is directed at verification
  - The abstraction might hide some flaws, but . . .
  - It is intended to make it easy to find common types of flaws
- This approach has been a remarkable success

Introduction

- Abstract protocol notation is directed at verification
  - The abstraction might hide some flaws, but . . .
  - It is intended to make it easy to find common types of flaws
- This approach has been a remarkable success

$\Rightarrow$ But there are a few weak spots . . .

A simple example

Send $n \in \mathbb{N}$ to B in secrecy using shared key k:

$$1.\ A \rightarrow B : \{n\}_k$$

Is this protocol secure?

A simple example

Send $n \in \mathbb{N}$ to B in secrecy using shared key k:

$$1.\ A \rightarrow B : \{n\}_k$$

Is this protocol secure?

Sure, it is secure under standard models.

A simple example

Send $n \in \mathbb{N}$ to B in secrecy using shared key k:

1. $A \to B : \{n\}_k$

Is this protocol secure?

Sure, it is secure under standard models.

But wait: it is insecure in practice if n is represented in unary!

A real-world attack

The SSH protocol (simplified):

1. $A \to B$: $\{\text{password}_A\}_k$

A real-world attack

The SSH protocol (simplified):

$$1.\ A \to B :\ \{\text{password}_A\}_k$$

Warning!

This protocol discloses the length of the password [SWT2001].

On message integrity

Send an authenticated message:

1. $A \rightarrow B : \{m\}_k$

Is this protocol secure?

On message integrity

Send an authenticated message:

1. $A \rightarrow B : \{m\}_k$

Is this protocol secure?

Sure, it is secure under standard models.

On message integrity

Send an authenticated message:

1. $A \rightarrow B: \{m\}_k$

Is this protocol secure?

Sure, it is secure under standard models.

But wait: If you implement $\{m\}_k$ as encryption, with no MAC, it may be insecure in practice. (Consider a stream cipher.)

On conditional operations

Send $b \in \{0, 1\}$ to B in secrecy:

$$1.\ A \to B : \{b\}_k$$

$$2.\ B \to A : \mathsf{ok} \qquad \text{(only if } b = 1\text{)}$$

Is this protocol secure?

On conditional operations

Send $b \in \{0, 1\}$ to B in secrecy:

1. $A \rightarrow B : \{b\}_k$
2. $B \rightarrow A :$ ok (only if $b = 1$)

Is this protocol secure?

No way. Presence/absence of message 2 reveals b.

A second real-world attack

The 802.11 WEP protocol (simplified):

$$\texttt{1.}\quad A \rightarrow B : \{\text{a TCP packet}\}_k$$

A second real-world attack

The 802.11 WEP protocol (simplified):

1. $A \to B$: $\{\text{a TCP packet}\}_k$

Warning!

Presence/absence of a TCP acknowledgement reveals one bit of the plaintext. Repeating the attack allows to use B as an oracle to decrypt the entire plaintext for us [BGW2001].

Conclusions

- Fixes:
  - Length attacks: Extend the formal model somehow?
  - Secrecy vs. integrity: Use $\{x\}_k$ for secrecy, $[x]_k$ for integrity
  - Conditional operations: Something like non-interference??
- Lessons:
  - Abstract protocol notation may hide real-world defects
  - Interactions between crypto & system can cause subtle flaws