Privacy and Security in Library RFID Issues, Practices, and Architectures

David Molnar & David Wagner

Presented by Andrew K. Adams

Motivation

Supply chain applications focus on *pallet* or *case* RFID tagging, but, library applications require *item-level* tagging.

"In an item-level tagging regime, the ability to track tags raises the possibility of surveillance of library patrons and their reading habits."

Once a library selects an RFID system, upgrade is unlikely.

Goal

"Propose a new architecture for using RFID technology securely in libraries without compromising privacy."

Hurdles:

- private authentication
- key management

RFID Background

Passive technology (only powered by reader)

Short-range (pre-computation impossible)

Not crypto-capable (few gates)

Proprietary (Checkpoint, TAGSYS)

ISO 15693 (Texas Instruments)

ISO 18000-3 (Modes 1 & 2)

- Mode 2 has RNG, higher speeds, etc.

Library RFID Architecture

Bibliographic database

- unique number (bar code)
- extras (location, title, author, check-out)

During check-out/check-in

- reader gets ID from tag
- status of ID in database changed

Library Architecture, Cont.

Security *features* with RFID

- reader (by exit) can (repeat) the database lookup on the book's status
- check-out reader can set the *security bit*, then exit reader can check that bit

You have to love when the primary purpose of a mechanism is *management*, but then someone goes and uses it for *security*!

- a violation of Security of Mechanism?

Current State (in 2004!)

Dogma

> "An adversary without access to the bibliographic database and with only short-range readers poses little to no risk."

Current State (in 2004!)

Dogma

> "An adversary without access to the bibliographic database and with only short-range readers poses little to no risk."

Question: Does this argument sound familiar regarding something being deployed in the last year? (I'm not talking about credit cards.)

Current State (in 2004!)

Dogma

> "An adversary without access to the bibliographic database and with only short-range readers poses little to no risk."

Question: Does this argument sound familiar regarding something being deployed in the last year? (I'm not talking about credit cards.)

Unique IDs, no read passwords and security bits throw a wrench in this tenet.

Attacks

Origin inference

unique IDs have geographic prefixes

Tracking

correlate readings of a specific ID

Hotlisting

adversary has a list of IDs in advance

"Look Out! He's got an almanac!"

http://cryptome.org/fbi-almanacs.htm

Collision-Avoidance

Because many RFIDs may be in range of a reader at the same time ...

ISO 18000-3 mode 1 (globally unique 64bits)

- respond to INVENTORY command
- also will respond to a variable-length mask that matches its ID

ISO 18000-3 mode 2 (64bit mfr ID)

- random number in collision avoidance
- most likely mfr ID will be seed

Impact

Oh, and it gets worse:

> "The collision-avoidance behavior is hard-coded at such a low layer of the tag that no matter what higher layers do, privacy will be unachievable."

Impact

Oh, and it gets worse:

> "The collision-avoidance behavior is hard-coded at such a low layer of the tag that no matter what higher layers do, privacy will be unachievable."

That is, even if we applied access control to prevent unauthorized reading of the tags, we're still hosed ...

... talk about a let down, we haven't even got to chapter 4 yet!

Tag Password Management

Assuming *private* collision-avoidance exists!

Single secret per-site

- a compromise of one results in a full compromise system

Each tag has different secret

- mechanism required to allow reader to tell what secret to use

Any serious security *dictates* separate secrets ...

Private RFID Mechanisms

Random Transaction IDs on Rewritable Tags

- during check-out, reader learns tag ID (in library!)
- reader generates random number (r)
- reader stores pair (ID, r) in database
- reader erases ID on tag
- reader inserts r on tag

Note, it solves origin inference & hotlisting.

Private RFID Mechanisms, Cont.

Improved Passwords Via Persistent State:

reader → HELLO → tag

reader ← r (nonce) ← tag

reader → (cmd, $\rho = s \oplus r$) → tag

Assumes tag → reader channel secure.

Requires good randomness at tag.

Metrics

"We will say a scheme is private if an adversary is unable to distinguish two different tags with different secret keys, and secure if an adversary cannot fool a tag or reader into accepting when it does not in fact know the secret key."

Note, *Improved Passwords Via Persistent State* is private, but not secure.

Additionally, we care how the amount of work at the reader scales with the number of tags.

Previous Work

Randomized Hash Lock Protocol (Weis et al.)

generate key, ID pairs, store in database

reader $\leftarrow$ $(r, f_s(r) \oplus ID)$ $\leftarrow$ tag

reader finds pair that satisfy $f_s(r) \oplus ID$

reader $\rightarrow$ ID $\rightarrow$ tag

Workload linear in regard to number of tags.

Neither private or secure, hmm ...

Stronger Mechanism

Basic PRF Private Authentication Scheme

reader $\rightarrow$ HELLO, r_1 $\rightarrow$ tag

reader $\leftarrow$ r_2, $\sigma = ID \oplus f_s(0,r_1,r_2)$ $\leftarrow$ tag

reader finds secret, ID pair in database

reader $\rightarrow$ $\tau = ID \oplus f_s(1,r_1,r_2)$ $\rightarrow$ tag

Workload linear in regard to number of tags.

Stronger Mechanism

Basic PRF Private Authentication Scheme

reader $\rightarrow$ HELLO, r_1 $\rightarrow$ tag

reader $\leftarrow$ r_2, $\sigma = ID \oplus f_s(0,r_1,r_2)$ $\leftarrow$ tag

reader finds secret, ID pair in database

reader $\rightarrow$ $\tau = ID \oplus f_s(1,r_1,r_2)$ $\rightarrow$ tag

Workload linear in regard to number of tags.

Question: Is this susceptible to ghost & leech?

A Scalable Mechanism

Tree-based Private Authentication:

- n-tags are leaves in balanced binary tree
- each edge assigned a secret
- tags contain $\log_2 n$ edge-secrets of path
- reader starts at root, tries both edges
- reader needs to succeed with one secret at each edge to continue towards tag's ID

Tree-based Algorithms

$G_{tree}(1^k, N)$

Fix l ← logN
for i = 1 to l
  for j = 0 to 1
    $s_{i,j}$ ← $G_1(1^k)$
for h = 1 to N
Parse h in binary as $(b_1, ..., b_l)$
TK_h $(s_{1,b1}, ..., s_{l,bl})$
RK $(s_{1,0}, s_{1,1}, ..., s_{l,1})$
Output R_K, TK_1, ..., TK_N.

(R_{tree}, T_{tree}) (RK, TK)

Fix l ← logN
Parse RK as $(u_{1,0}, u_{1,1}, ..., u_{l,1})$
Parse TK as $(v_1, ..., v_l)$
for i = 1 to l
  succeed ← false
  for j = 0 to 1
    if running $(R1(u_{i,j}); T1(v_i))$ returns true
    then succeed ← true
  if ¬succeed
  then fail and output 0
accept and output 1

Tree-based Example

Example (n = 16, so l = 4):

Generator generates 8 secret keys:

RK = $s_{1,0}$, $s_{1,1}$, $s_{2,0}$, $s_{2,1}$, $s_{3,0}$, $s_{3,1}$, $s_{4,0}$ $s_{4,1}$

Tag_3 (in 2^l binary) = 0011

Thus, TK_3 gets keys: $s_{1,1}$, $s_{2,1}$, $s_{3,0}$, $s_{4,0}$

So, reader tries $S_{1,0}$ & $S_{1,1}$ at first level

$S_{1,1}$ succeeds, so reader tries $S_{2,0}$, $S_{2,1}$ at second level ... so on and so forth.

Tree-based Performance

Tree-based scheme *can* use Basic Private Authentication Scheme ...

O(log n) work for reader

O(k log n) communication cost

O(log n) storage at tag

It may be that O(k log n) is too much communication cost, so ...

More-efficient Mechanism

Two-Phase Tree Scheme:

- phase 1, use tree-based scheme to learn tag's ID
- phase 2, command issued to tag ID
- in phase 1, PRF (i.e., $f_s(0,r_1,r_2)$) is truncated to a much smaller value
- phase 2 uses full security parameter k with PRF
- thus, communication cost is O(k + log n)

Strength & Weaknesses

I liked that:

Tree-based scheme is parallelizable.

The authors recognize the potential dangers of hotlisting.

Let's face it, authenticating via a key-path was pretty cool!

I disliked that:

Not even an attempt to solve *collision-avoidance privacy*!

Questions?

Rewritable tags