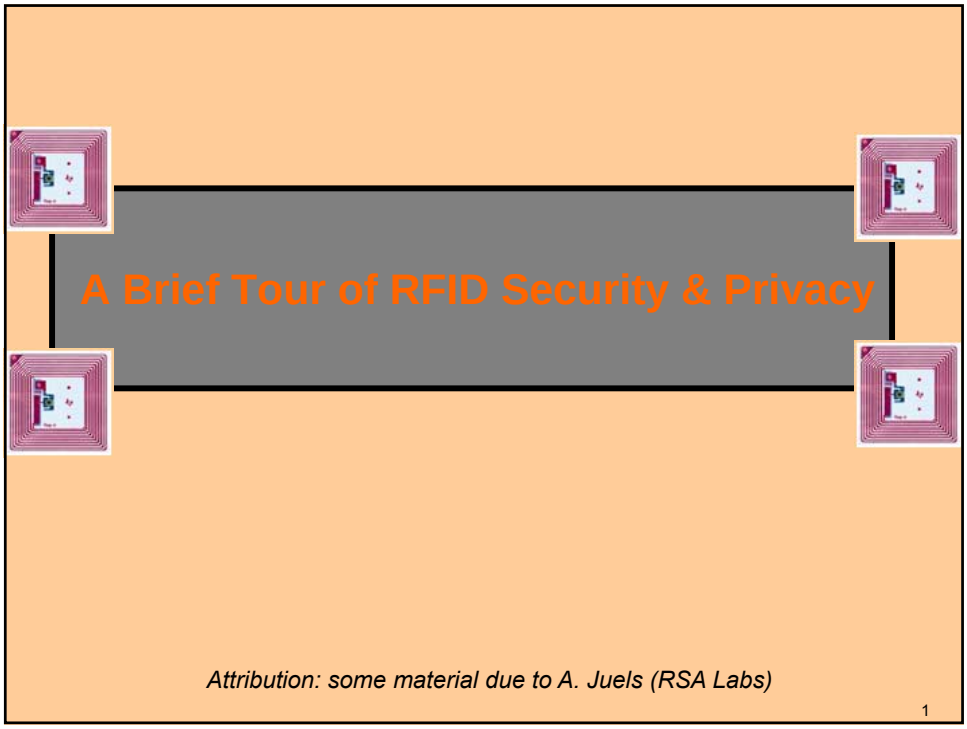
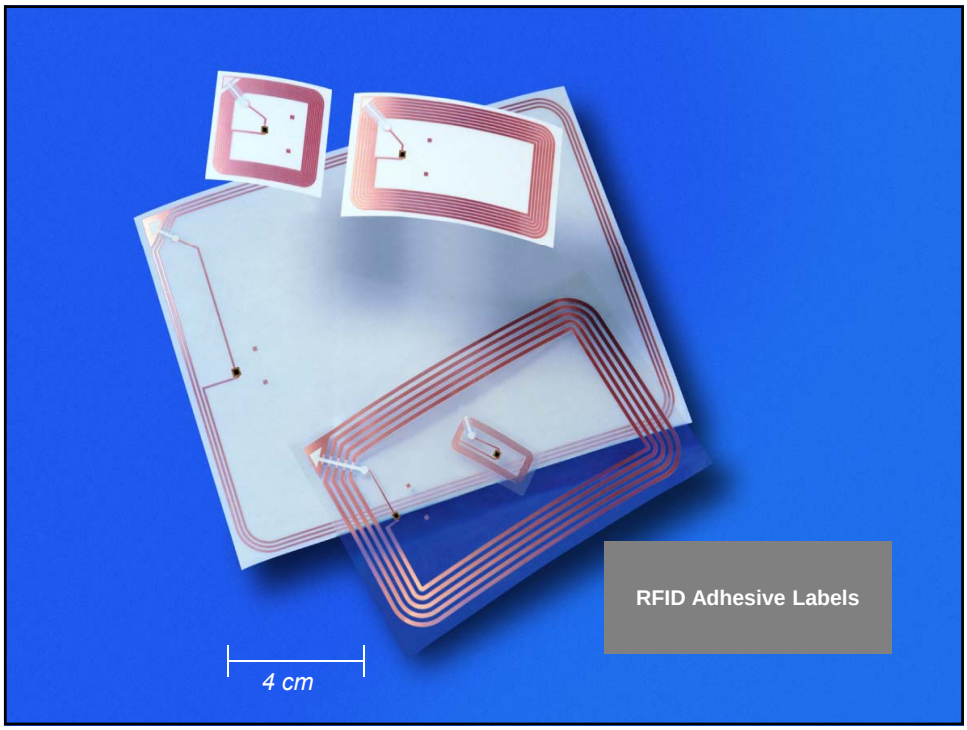

A Brief Tour of RFID Security & Privacy
Attribution: some material due to A. Juels (RSA Labs)
1


RFID Adhesive Labels
4 cm

## RFID System

Components of an RFID System:

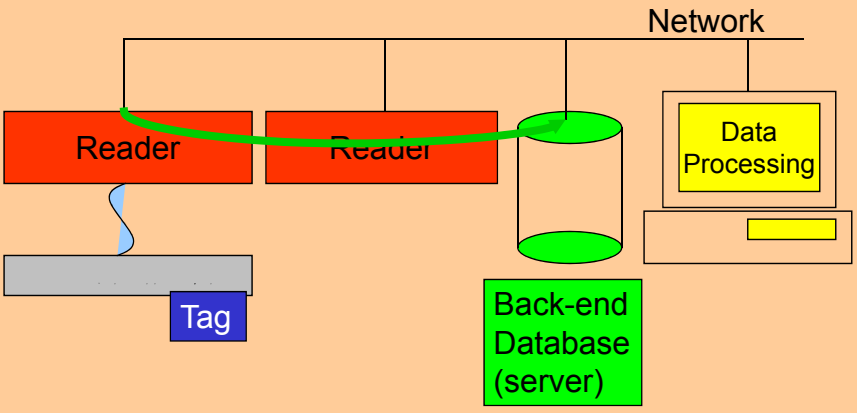
- Tags (transponders):
  - affixed to objects and carry identifying data.

- Readers (transceivers):
  - read or write tag data and interface with back-end databases

- Back-end databases (servers):
  - correlate tag data with objects

3

## System Interface




4

## RFID History

- Earliest Patent: John Logie Baird (1926)
- “Identify Friend or Foe” (IFF) systems developed by the British RAF to identify friendly aircraft.
- Both sides secretly tracked their enemy’s IFF.
- How do you identify yourself only to your friends?


5

## Commercial Applications

- Early Applications:
  - Tracking boxcars and shipping containers.
  - Cows: RFID ear tags.
  - Bulky, rugged, and expensive devices.
- The RFID Killer App?
  - Replace bar codes!


6

## Supply-Chain Management

- First Universal Product Code (UPC) scanned: a pack of Juicy Fruit gum in 1976.
- Every day, over 5,000,000,000 barcodes are scanned around the world.
- Barcodes are slow, need line of sight, physical alignment, and take up packaging "real estate"
- Over one billion RFID tags on the market.

7

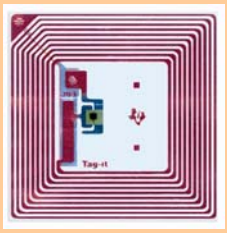
## Modern RFID Applications

- Supply-Chain Management
  - Inventory Control
  - Logistics
  - Retail Check-Out
- Access Control: Facility Access Proximity Cards (contact-less badges / smartcards)
- Payment Systems: Mobil SpeedPass.
- Medical Records
- Pet tracking chips

8

## Many "faces" of RFID devices





9

## Variety of RFID Technologies

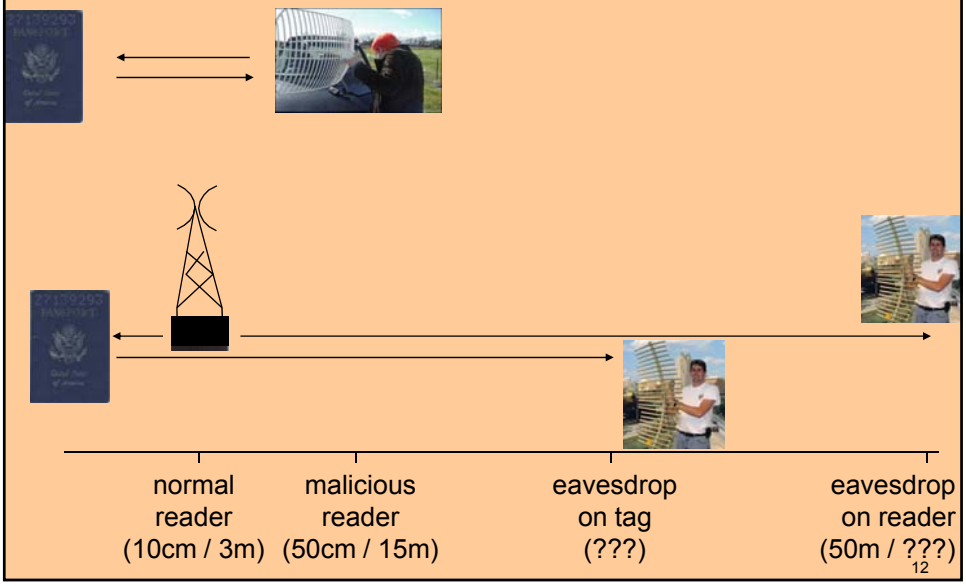

10

## Tag Power Source

- Passive (true RFID):
  - All power comes from a reader's interrogation signal
  - Tag is inactive unless a reader activates it
  - Passive powering is the cheapest; but short range
- Semi-Passive (more like a sensor) :
  - Tags have an on-board power source (battery).
  - Cannot initiate communications, but can be sensors.
  - Longer read range, more cost for battery.
- Active (more like a "fancy" sensor or PDA):
  - On-board power and can initiate communications.

11

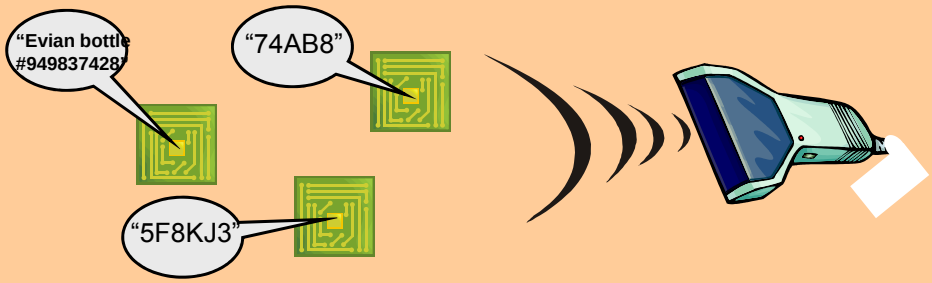
## READ RANGE?




12

## "Smart label" RFID tag

- Passive device – receives power from reader
- Range of up to several meters
- Simply calls out (unique) name and static data




13

## Capabilities of "smart label" RFID tag

- Very little memory
  - Static 96-bit+ identifier in current ultra-cheap tags
  - Hundreds of bits soon
- Little computational power
  - Several thousand gates (mostly for basic functionality)
  - ***No* real cryptographic functions possible**
  - Pricing pressure may keep it this way for a while

14

## What the future has in store for us: EPC (Electronic Product Code) tags

Barcode

EPC tag

***Fast, automated scanning***

**Line-of-sight** — **Radio contact**

**Specifies object type** — **Uniquely specifies object**

***Provides pointer to database entry for every object, i.e., unique, detailed history***

Not just object type/class!

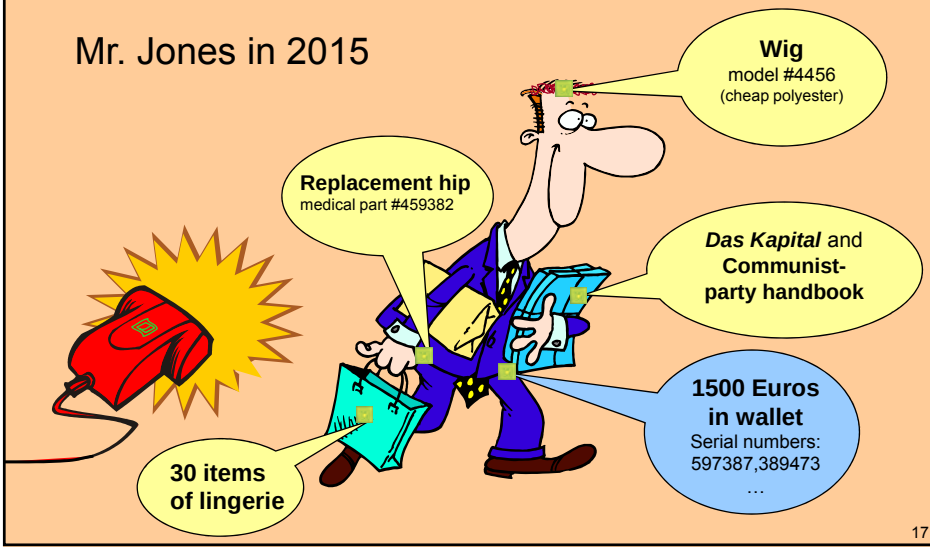
15

## So, what are the problems?

16

## The privacy problem

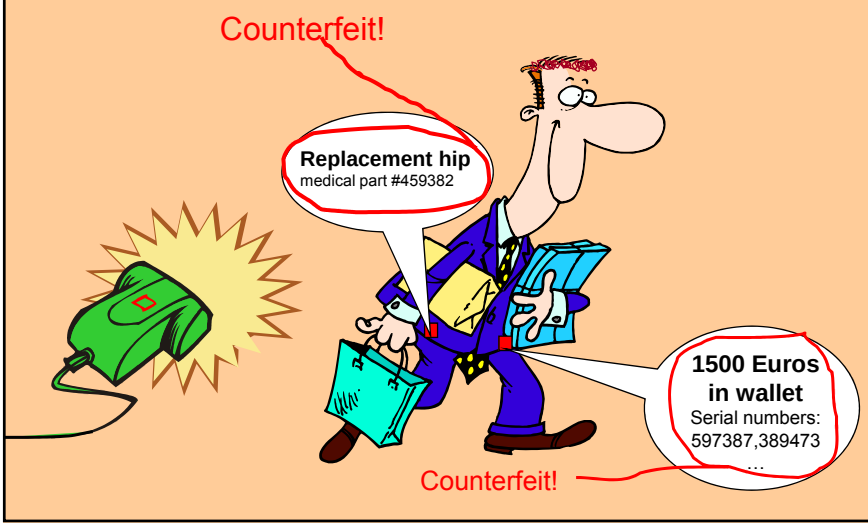
Bad readers, good tags




17

## The authentication problem

Good readers, bad tags




18

## Security Risks: Espionage/Privacy

- Espionage:
  - Identify Valuable Items to Steal
  - Monitor Changes in Inventory
- Personal Privacy
  - Leakage of personal information (prescriptions, brand/size of underwear, etc.).
  - Location privacy: tracking physical location of individuals by their RFID tags.

19

## Example

- The US Food and Drug Administration (FDA) recommended tagging prescription drugs with RFID "pedigrees".
- Problems:
  - "I'm Morphine. Steal me!"
  - "Bob's Viagra bottle is empty…"
  - "Hi. I'm Alice's anti-herpes cream."

20

## Security Risks: Forgery

- RFID casino chips, Mobil SpeedPass, EZ-Pass, FasTrak, prox cards, €500 banknotes, designer clothing.
- Skimming: Read your tag, make my own.
- Swapping: Replace real tags with decoys.
- Producing a basic RFID device is simple.
  - A “hobbyist” hacker can probably spoof most RFID devices in a weekend for under $50.

21

## Security Risks: Sabotage

- If adversary can’t eavesdrop or forge valid tags, can simply attack the RFID infrastructure.
  - Erase inventory data.
  - Vandalize – “kill” tags by demagnetizing

22

## Security Challenge

- Resources, resources, resources…
- EPC tags ~ 5 cents. 1000 gates ~ 1 cent.
- Main security challenges come from resource constraints.
- Gate count, memory, storage, power, time, bandwidth, performance, die space, and physical size are all tightly constrained.
- Pervasiveness (scale) also makes security hard.

23

## Example Tag Specification

| | |
|---|---|
| **Storage** | 128-512 bits of read-only storage. |
| **Memory** | 32-128 bits of volatile read-write memory. |
| **Gate Count** | 1000-10000 gates |
| **Security Gate Budget** | 200-2000 gates. |
| **Operating Frequency** | UHF 868-956 MHz. |
| **Forward Range** | 100 meters. |
| **Backward Range** | 3 meters. |
| **Read Performance** | 100 read operations per second. |
| **Cycles per Read** | 10,000 clock cycles. |
| **Tag Power Source** | Passively powered via RF signal. |
| **Power Consumption per Read** | 10 µWatts |
| **Features** | Anti-Collision Support<br>Random Number Generator (from outside) |

24

## Resource Constraints

- With such constraints, modular-math-based public-key algorithms like RSA or ElGamal are **much** too expensive
- Alternative public-key cryptosystems like ECC, NTRU, or XTR are also too expensive
- Even symmetric encryption is too costly.

  Can't fit DES, AES, or SHA-1 in 2000 gates

  But, recent progress made with AES, see:

  L. Batina, et al.

  "Public-Key Cryptography for RFID-Tags"

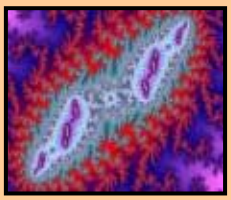
  PerCom Workshops 2007.

25

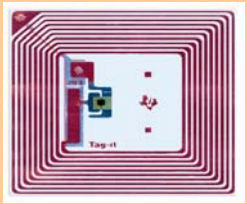
## RFID security challenge

How to obtain maximum security & privacy with minimal resources?

An RFID tag is a computational Amoeba



=



26

## Let's take a look at HOW NOT TO DO IT RIGHT

27

## The Digital Signature Transponder

A. Juels, S. Bono, M. Green, A. Stubblefield, A. Rubin, and M. Szydlo
USENIX Security '05


"I'm tag #123"

40-bit challenge $C$

24-bit response $R = f_K(C)$

(simplified)


- Helps secure tens of millions of automobiles
  - Philips claims more than 90% reduction in car theft thanks to RFID! (TI did at one point.)
- Also used in millions of payment transponders

28

## The Digital Signature Transponder (DST)


"I'm tag #123"

40-bit challenge $C$

24-bit response $R = f_K(C)$

(simplified)


- The key $K$ is only 40 bits in length!

29

## The Digital Signature Transponder (DST)


"I'm tag #123"

40-bit challenge $C$

24-bit response $R = f_K(C)$

(simplified)


**Goal: Demonstrate security vulnerability by cloning real DST keys**

30

## The Digital Signature Transponder (DST)


"I'm tag #123"

40-bit challenge $C$

24-bit response $R = f_K(C)$

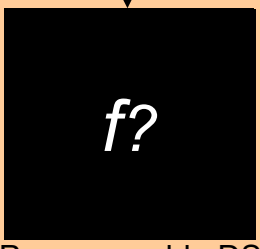
(simplified)


- The key $K$ is only 40 bits in length!
- But what is the cryptographic function $f$?

31

## Black-box cryptanalysis

key $K$




Programmable DST

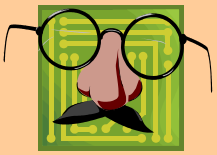
$C$

$R = f_K(C)$

32

## The full cloning process

1. Skimming
2. Key cracking
3. Simulation



33

## The full cloning process

**Step 1: Skimming**

Obtain responses $r_1, r_2$ to two challenges, $c_1, c_2$

(1/4 second)

34

## The full cloning process

### Step 2: Key cracking

Find secret key $k$ such that $r_1=f_k(c_1)$ and $r_2 = f_k(c_2)$

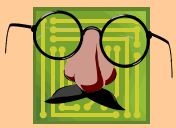
(30 mins. on 16-way parallel cracker)

35

## The full cloning process

### Step 3: Simulation

Simulate radio protocols with computation of $f_k$



36

## Problems Re-cap

- Privacy, i.e., tracking tags by:
  - Eavesdropping on tag←→reader interaction
  - Rogue readers interrogating tags
  - Identifying product-line (merchandise type)
- Security:
  - Tag cloning / impersonation
- Denial-Of-Service:
  - Killing / incapacitating tags

37

## Solutions?

- Encryption (randomized): against eavesdropping (tracking)
- Tag → reader authentication: against cloning & counterfeiting
- Reader → tag authentication: against rogue readers (tracking)
- Tamper-resistance: against tracking and cloning (expensive for very cheap tags…)

38

## Caveats:

However:

- Ideally, no more than 2 messages in reader-interaction; tags can't keep temporary "state"
- Tag can't challenge reader: good randomness hard to obtain – true RNGs not cheap
- Ideally, no more than 1 simple crypto operation on tag (e.g., keyed hash)
- Can't have one key for all tags: **one attack** pays off very well
- Can't have one key for all reader: same reason as above
- Can't make reader do on-the-fly O(n) computation where n = total # of tags -- n can be VERY large

Ideally would use group signatures + secret handshakes

but cost prohibits it…

39

## "Batch" vs "Interactive" Mode

- Interactive
  - Tag scanned by reader and immediately identified/authenticated
  - Either reader "knows" all tags or connects in real time to back-end server
- Batch
  - Reader scans a multitude of tags
  - Obtain responses
  - At later time, batches responses over to back-end server
  - Server identifies/authenticates tags

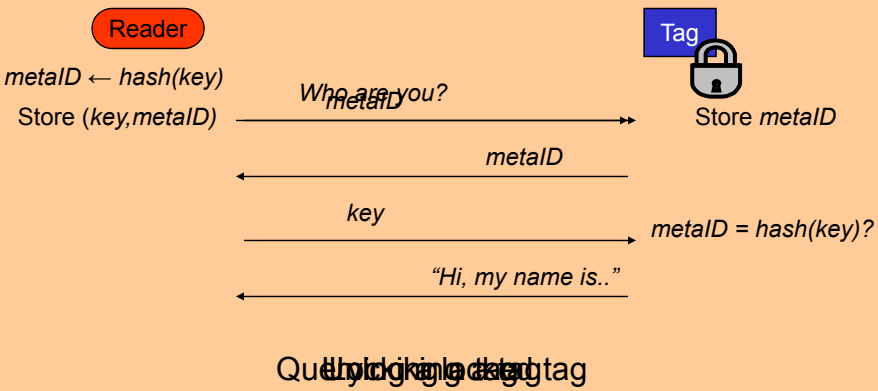
40

## Hash Locks

- Rivest, Weis, Sarma, Engels (2003).
- Access control mechanism:
  - Authenticates readers to tags.
- "Only" requires OW hash function on tag.
- Lock tags with a one-way hash output.
- Unlock tags with the hash pre-image.
- Old idea, new application.

41

## Hash Lock Access Control



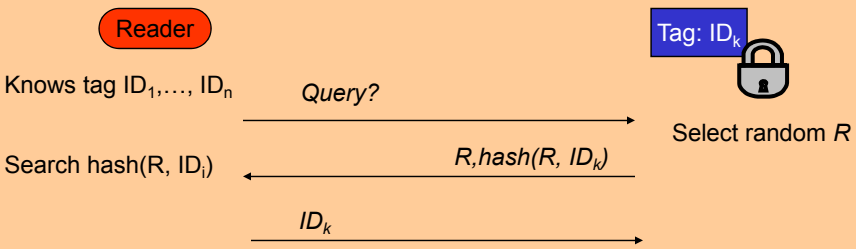

42

## Hash Lock Analysis

+ Cheap to implement on tags:
  A hash function and storage for *metaID*.
+ Security based on hardness of hash.
+ Hash output has nice random properties.
+ Low key look-up overhead.
- Tags respond predictably; allows tracking.
  Motivates randomization.
- Too many messages/rounds
- Requires reader to know all keys

43

## Randomized Hash Lock




Unlocking a tag

44

## Randomized Hash Lock Analysis

+ Implementation requires hash and random number generator
  - Low-cost PRNG.
  - Physical randomness.
+ Randomized response prevents tracking.
- Inefficient brute force key look-up.
- Hash only guaranteed to be one-way. Might leak information about the ID.

  *(Essentially end up with a block cipher?)*

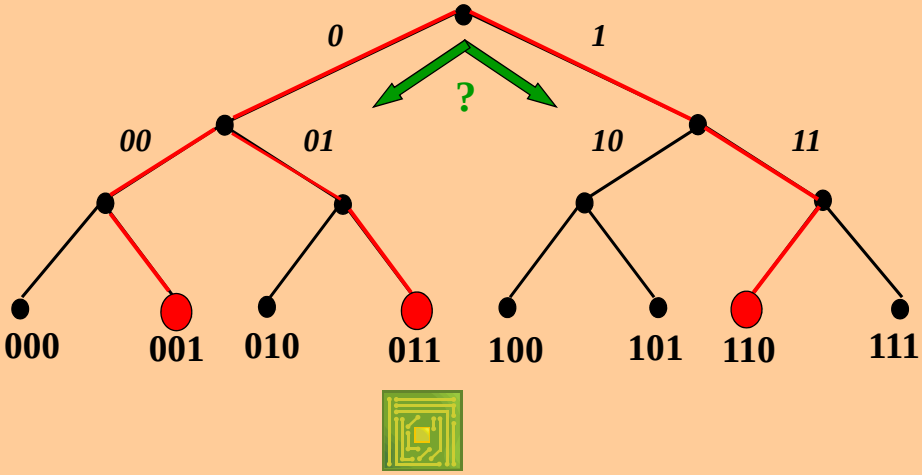
45

## Blocker Tags

- Juels, Rivest, Szydlo (2003).
- Consumer Privacy Protecting Device:
  - Hides your tag data from strangers.
- Users carry a “blocker tag” device.
- Blocker tag injects itself into the tag’s anti-collision protocol.
- Effectively spoofs non-existent tags.
- Concept *only exists on paper.*

46

## "Tree-walking" anti-collision protocol for RFID tags




47

## In summary:

- "Tree-walking" protocol for identifying tags recursively asks question:
  - "What is your next bit?"
- Blocker tag always says ***both '0' and '1'***!
  - Makes it seem like *all* possible tags are present
  - Reader cannot figure out which tags are actually present
  - Number of possible tags is *huge* (at least a billion billion), so reader stalls

48

## Noisy Tags: Eavesdropping Protection

C. Castelluccia and G. Avoine, CARDIS 2006

- Eavesdropping (passive attacks) can be prevented by encrypting data between the tag and the reader…
- This requires establishing a key…
- Current key exchange solutions are too expensive for cheapest RFID tags
  - Very little memory
    - Static 96-bit+ identifier in current ultra-cheap tags
    - Hundreds of bits soon
  - Little computational power
    - Several thousand gates (mostly for basic functionality)
  - Pricing pressure may keep it this way for a while
- Need a way for a tag to confidentially send its ID to the reader, without any computation

49

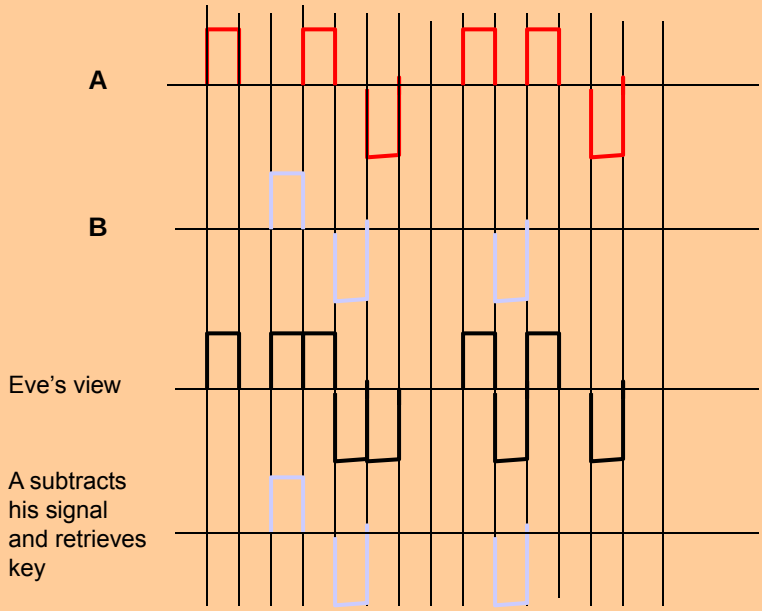
## Basic Idea:
### How to send a secret without computing

Based on an idea proposed by Bell Labs a few decades ago…

- A and B want to share a secret key
- A sends some random signal on the channel
- B sends simultaneously the secret on the channel
- A removes the noise and retrieve the secret
- An eavesdropper, Eve, only sees noise and cannot retrieve the key…

50

## Assuming random access to the channel (CSMA)…




51

## Application to RFID…

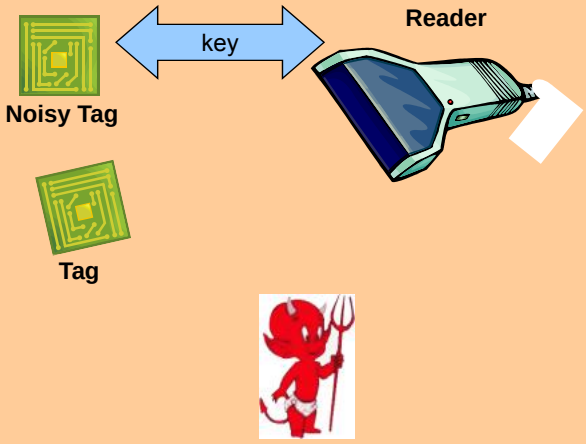
A noisy tag:

- a regular tag (in the reader "environment" ) which generates "noise"

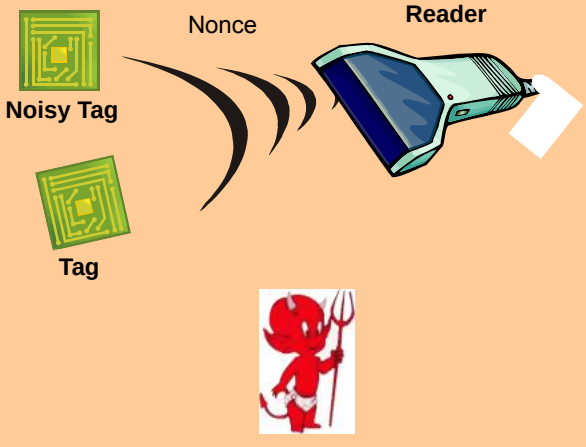
- Noisy tag shares a secret key with the reader
- Noisy tag reply generated from secret key ➔ can be predicted by reader
  - i.e. reply = hash(key, nonce)
- When reader queries a regular tag, it gets 2 bits back:
  - One from the noisy tag that it can compute and cancels out
  - One from the tag that is the secret bit
- Eve sees 2 bits and does not know which bit was sent by the tag!!!
  - Only works if the 2 bits are different
  - If bits are same, the round must be ignored
- An n-bit key can be exchanged after, on average, **2n rounds**.

52

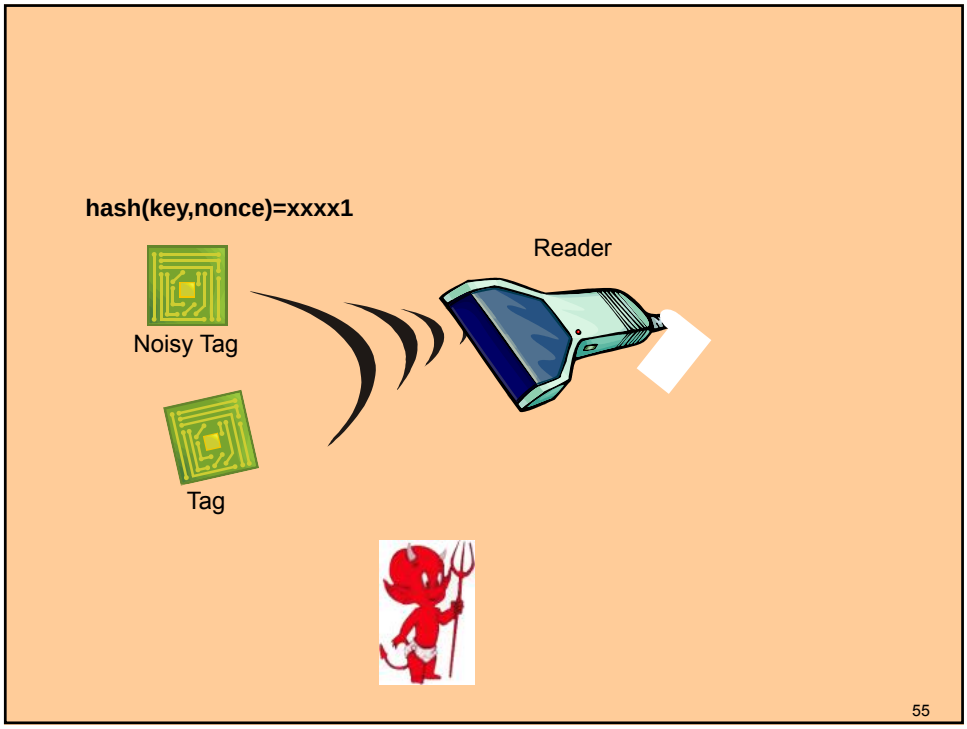
## Bit-based Scheme

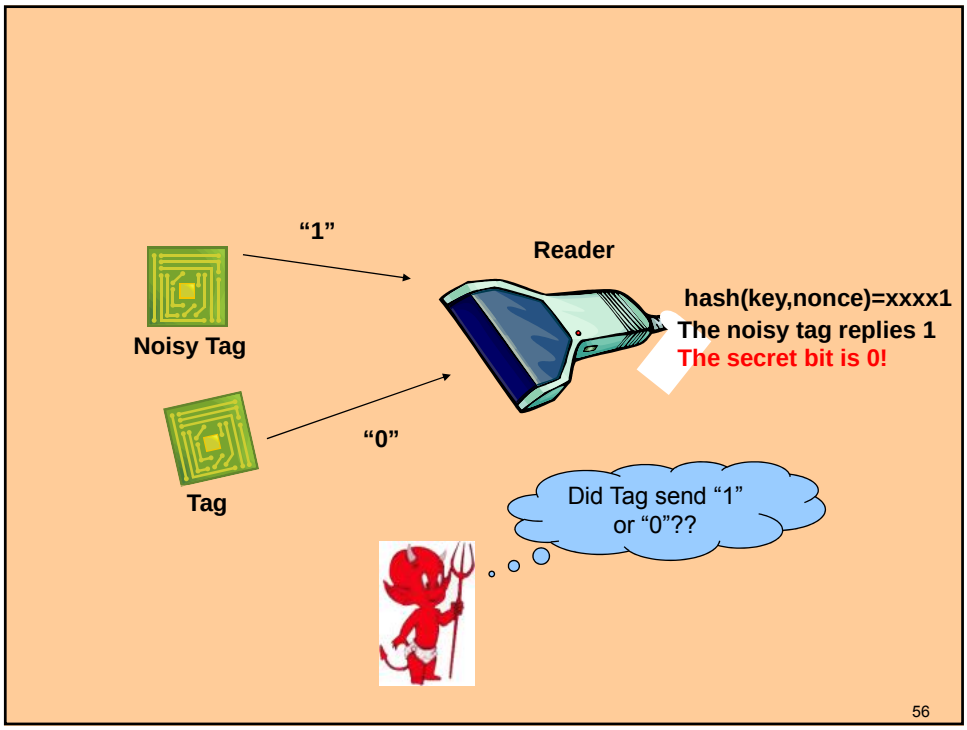



53




54


hash(key,nonce)=xxxx1
Reader
Noisy Tag
Tag
55


"1"
Reader
hash(key,nonce)=xxxx1
The noisy tag replies 1
The secret bit is 0!
Noisy Tag
"0"
Tag
Did Tag send "1" or "0"??
56

## Security

- Assuming that:
  - Bits sent by noisy tags are uniformly distributed
  - Bits sent by regular tags are uniformly distributed
  - Adversary can't determine (with a prob. > than ½) the source of a signal

=> The scheme is perfectly secure

57

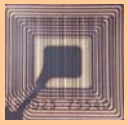
## How to Reduce Reader Computation?

- Molnar, et al.
  "Privacy For RFID Through Trusted Computing"
  WPES 2005.

- Molnar, et al.,
  "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags"
  SAC 2005.

58

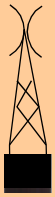
## A first attempt at defeating eavesdropping and unauthorized tag-reading



k

$E_k(r, ID)$

"pseudonym"



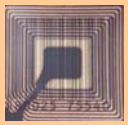
k

Decrypt & identify

Problem:
- All tags and readers share the same key k
- If any tag is compromised, all security is lost
- If any reader is compromised, all security is lost
- No authentication

59

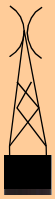
## Another extreme: uniquely-keyed tags



$k_i$

$r, F_{ki}(r)$

"pseudonym"



$(k_1, ID_1)$
:
$(k_N, ID_N)$

Scan, re-compute & identify

Problem:
- Doesn't scale
- Takes O(N) work to decode each pseudonym
- No authentication

60

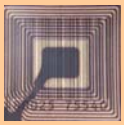
## Private identification protocols

Goal: a tag-reader protocol, providing:
- Identification: Authorized reader learns tag's identity
- Privacy: Unauthorized readers learn nothing
  - Attacker shouldn't even link two "sightings" of same tag
- Authentication: Tag identity cannot be spoofed
- Scalability: Can be used with many tags

A real technical challenge

61

## Hierarchical private tag identification



$k_i$, $k_{ij}$

$r, F_{k_i}(r), F_{k_{ij}}(r)$

pseudonym

:
$(k_i, i)$
:
$(i, k_{ij}, ID_{ij})$
:

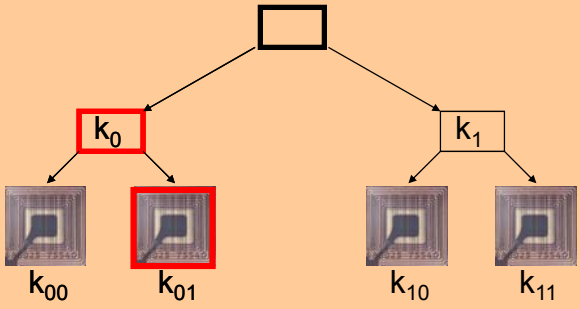
Decodes i, then j

More scalable: $O(\sqrt{N})$ work to decode each pseudonym
- First, try all $k_i$ to learn i
- Then, try all $k_{ij}$ to learn j and, thus, tag's real identity

BUT:
- Learning $k_i$ allows tracking the entire "family" of tags
- So, breaking into one tag allows family tracking!

62

## Tree of secrets (LKH)




- Tag ≡ leaf of the tree
- Each tag receives the keys on path from leaf to the root
- Tag ij generates pseudonyms as: $(r, F_{k_i}(r), F_{k_{ij}}(r))$
- Reader can decode pseudonym using depth-first search

63

## Analysis: tree of secrets

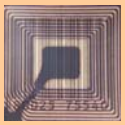
Generalizations:
- Use any depth tree (e.g., lg N)
- Use any branching factor (e.g., $2^{10}$)
- Use any other identification scheme (e.g., mutual auth)

| | Theory | Concrete example |
|---|---|---|
| Number of tags: | N | $2^{20}$ tags |
| Tag storage: | O(lg N) | 128 bits |
| Tag work: | O(lg N) | 2 PRF invocations |
| Communications: | O(lg N) | 138 bits |
| Reader work: | O(lg N) | $2 \times 2^{10}$ PRF invocations |

Privacy degrades "gracefully" if tags are compromised

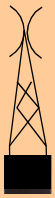
64

# Reducing trust in readers



$k_i$, $k_{ij}$

$r, F_{k_i}(r), F_{k_{ij}}(r)$



Reader

$r, F_{k_i}(r), F_{k_{ij}}(r)$

$ID_{ij}$

Trusted Center

… $(k_{ij}, Policy_{ij})$ …

- If readers are online, Trusted Center can do decoding for them, and enforce a privacy policy for each tag.
- No keys stored at reader ==> less chance of privacy spills.

65

# A Lightweight RFID Protocol to protect against Traceability and Cloning attacks

T. Dimitriou, IEEE Securecomm 2005

TAG | Reader | Trusted Center

$Request, N_R$

$h(ID_i), N_T, h_{ID_i}(N_T, N_R)$

$h(ID_i), N_T, h_{ID_i}(N_T, N_R)$

**Lookup in table, verify and update $ID_i$ to $ID_{i+1}$**

$h_{ID_{i+1}}(N_T, N_R)$

$h_{ID_{i+1}}(N_T, N_R)$

- simple, forward-secure
- 3 messages: undesirable → tag must keep transient state
- need PRNG
- robustness is a problem: 3rd message might not arrive…or arrive a year later!
- what if malicious reader wants to track tags?
- cannot batch communication with center

66

## YA-TRIP (Percom'06+PET'07)

- Efficient tag identification – avoids O(n) work by reader/server
- Tag->reader authentication (optionally, reader->tag)
- Batch applications: reader "talks to" many tags, identifies them later (at server)

- Each tag has single unique key -- $k_i$
- Each tag maintains monotonically increasing timer (counter) – $t_i$

67

## Time-Based "Identity"

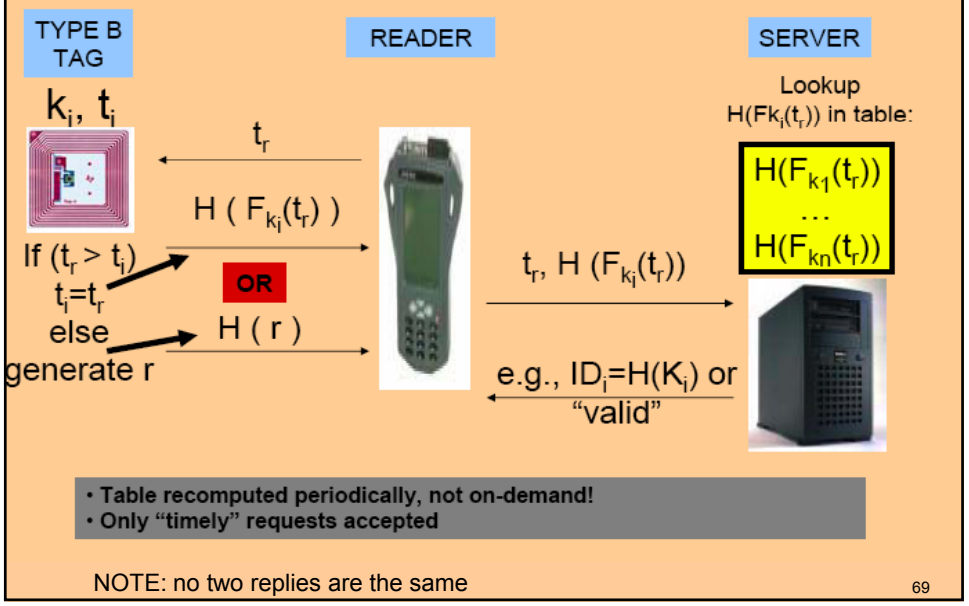
- Time-based table computed periodically (asynchronously) by the server
- For each time interval value $t_r$
- All readers are loosely time synchronized

Table for time $t_r$

| | |
|---|---|
| 1 | $H(F_{k1}(t_r))$ |
| | … |
| i | $H(F_{ki}(t_r))$ |
| | … |
| n | $H(F_{kn}(t_r))$ |

68

## YA-TRIP: Tag Identification

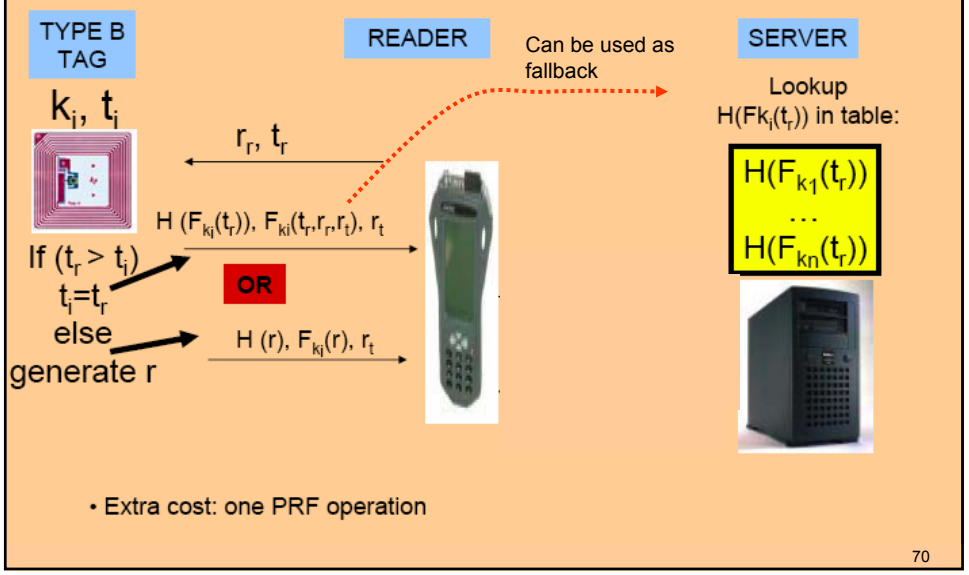



- Table recomputed periodically, not on-demand!
- Only "timely" requests accepted

NOTE: no two replies are the same

69

## YA-TRAP+: Tag Authentication

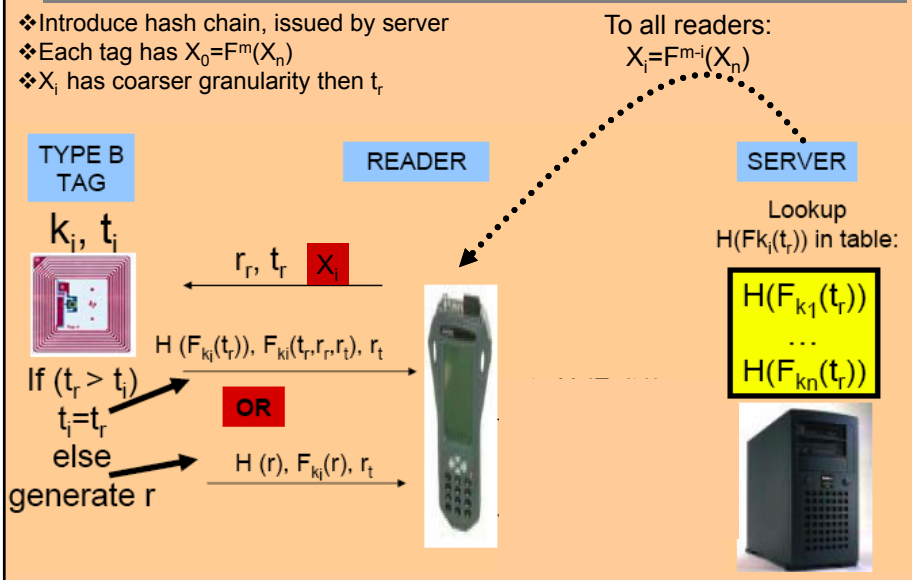



- Extra cost: one PRF operation

70

## YA-TRAP*: Reader Authentication
(aka DoS prevention)

- Introduce hash chain, issued by server
- Each tag has $X_0=F^m(X_n)$
- $X_i$ has coarser granularity then $t_r$




71

## Other extensions

- Can be combined with Molnar, at al. method (tree of secrets)
- Can be made forward-secure -- key evolves after each use
  - Breaking into a tag doesn't allow tracking tag's prior occurrences

72

## To learn more:

- Excellent Bibliography:
  - http://lasecwww.epfl.ch/~gavoine/rfid/
- Limited Bibliography:
  - crypto.csail.mit.edu/~sweis/rfid
- Primers and current RFID news:
  - www.rfidjournal.com
- RSA Labs RFID Web site:
  - www.rsasecurity.com/go/rfid
  - www.rfid-security.com
- JHU/RSA RFID Web site:
  - www.rfidanalysis.org
- David Wagner's Web site:
  - www.cs.berkeley.edu/~daw/papers

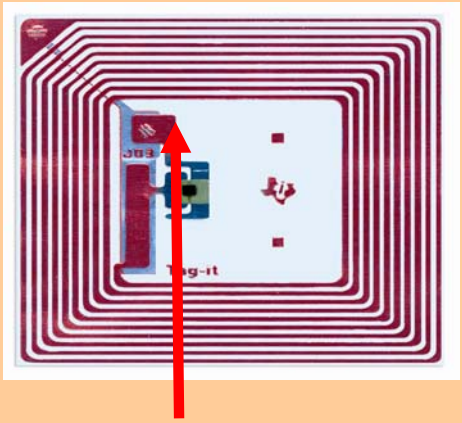
73

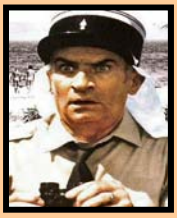
## RFID acceptance?

- Ultimately depends on the human user
- How to convince an average user that s/he has control over RFID tags
  - different from smartcards, tokens and PDAs
- Measures must be:
  - Human-assisted
  - Meaningful (e.g., visual)
  - Simple
  - Inexpensive
  - e.g., "Search and Destroy"

74

# RFID acceptance?





For example, use a toothpick-like piece of plastic to separate chip from antenna

Also, see:
G. Karjoth and P. Moskowitz,
Disabling RFID tags with visible confirmation: clipped tags are silenced.
WPES 2005

75