

Information Security Group
Royal Holloway, University of London

Improving RFID Protocol Security with Physics

Dr Gerhard Hancke

Where are we headed...

- RFID technologies used to be

- Reader and passive tag
- Single application
- Closed system
- Low value

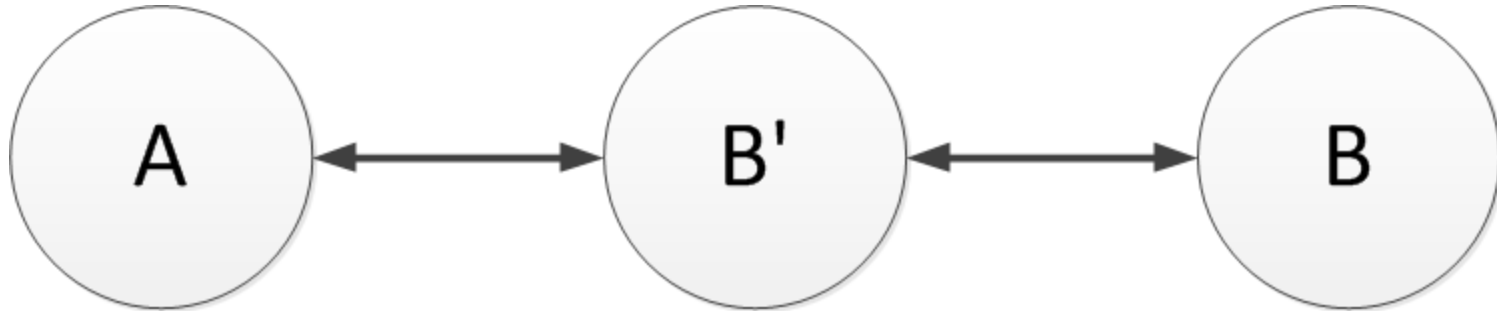


- Now a core 'Internet of Things' technology

- Definition of IoT is a bit vague, however....
- Devices need to communicate with each other
- Devices may have no/limited prior relationship

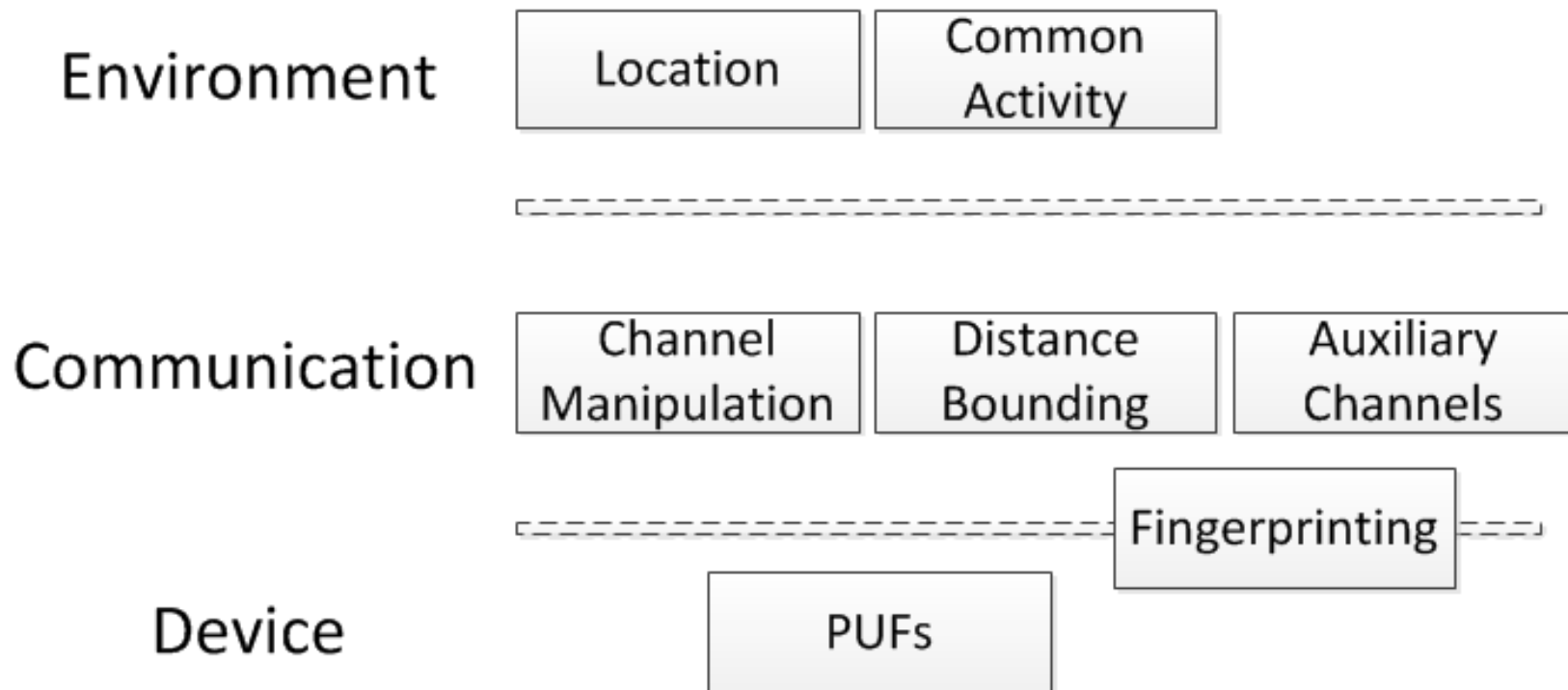


Some security issues



- Communication neighbour not physical neighbour
 - Is this who/what it claims it is?
 - MITM, mafia fraud, wormhole, relay...
- Security Requirements
 - Basic: Strong data origin authentication, proof of proximity
 - Messages bound to device, not only a key
 - 3 ➤ Advanced: Ad-hoc pairing/key agreement

Overview



- Physical observations part of human trust relationship
 - Physical characteristics or proximity of the other party
 - Incorporate physical context into security protocols

Physical Location

- Idea has been around for a while?
 - Location-based access control
 - Geographic packet 'leashes'
 - Current mobile devices to make it more feasible
- Approaches
 - Absolute location
 - Relative location
 - 'Dongle'
- Using location information
 - Proof of proximity
 - Pairing/key agreement

Auxiliary channel/Out-of-band

- Wireless communication
 - Difficult to determine actual source
 - Relatively easy to eavesdrop
- Use 'location limited' channels to set up communication
 - Audio
 - IR
 - Visual
- Using auxiliary channels
 - Proof of proximity (raises complexity of attack)
 - Pairing/key agreement

Common Activity

- Mobile devices have sensors
 - Accelerometer, gyroscope, temperature, etc
 - Verify that their physical context is the same
- Devices are set a physical task
 - Both devices take part in activity
 - Take measurements during activity
 - Show that they were physically involved
 - Practically limited to small, handheld devices
- Using physical activity
 - Proof of proximity
 - Pairing/key agreement



Distance-Bounding Protocols

- Physical distance
 - Device proves proximity to a verifying device
 - This communicating device is within x meters
- Use round trip time to estimate distance
 - Devices can execute protocol without external help
 - Special cryptographic challenge-response construction
 - Limit processing delay of the prover
 - Security and accuracy depends on the channel/hardware
 - Practical channel issue still unresolved
 - Consider practical and theoretical attack implementation
- Using distance bounding
 - Proof of proximity

Channel Manipulation

- RFID/contactless channel
 - Tags do not 'transmit' – it varies amplitude of reader's carrier
- The 'noisy' reader
 - Intentional noise or bit collisions
 - Third party cannot recover data but reader can
 - reader can remove noise, third party cannot
 - This process is transparent to tag (no shared key needed)
- Using channel manipulation
 - Privacy, confidentiality
 - Pairing/key agreement

Physically Unclonable Function (PUF)

- **Function intrinsically linked to device**
 - could be described as physical/hardware one way function
 - challenge-response authentication without key
 - Only specific, physical device can generate response

- **Good for intended function**
 - PUF prevents duplication of devices
 - infeasible to construct a PUF with chosen challenge-response behaviour
 - Practical and being implemented

- **Not really solving our problem....**
 - Still challenge-response that can be relayed

Device Fingerprinting

- Linked to the physical device
 - Variations in communication channel implementation
 - Features in time and frequency domains
 - Detects if communication is relayed

- Nice idea, but...
 - Still proof of concept – works in labs for UHF and HF devices
 - Special (complex) hardware to verify feature
 - Feature stability in RFID environment an issue
 - noise, tag position/orientation, etc
 - Feature entropy is low - mitigated by tendency to group tags...

Tags generally come in groups....

The consumer privacy problem

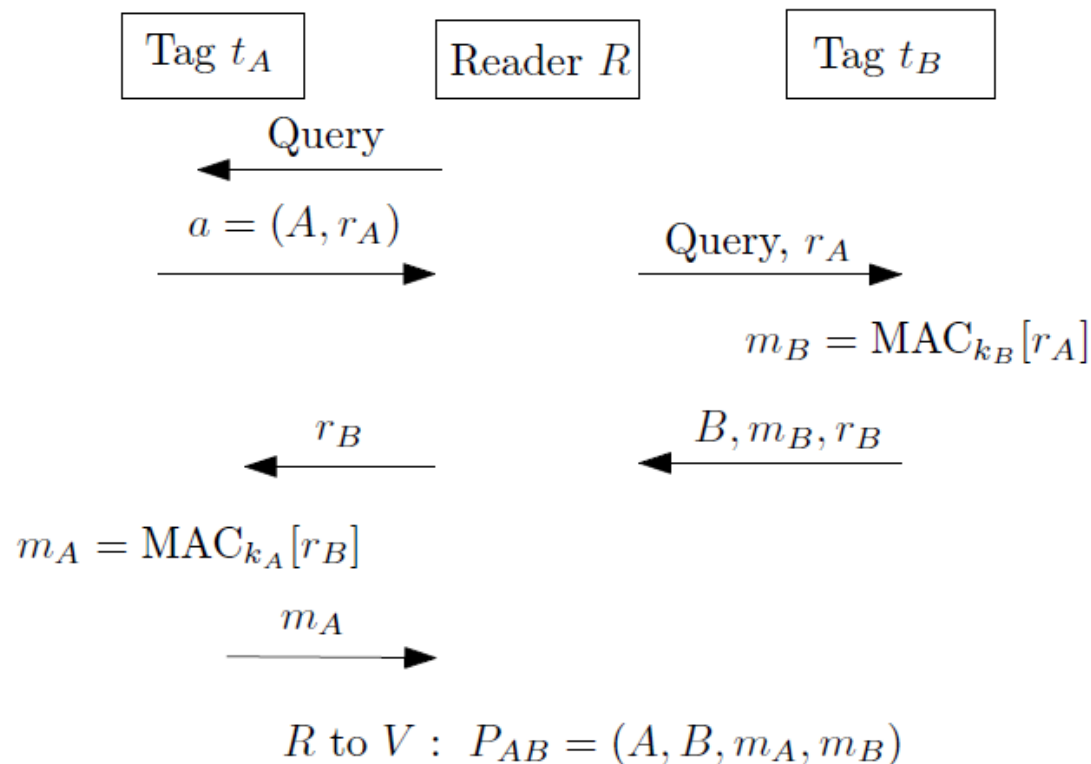


- Which brings us to second part of the talk....

RFID operation and groups of tags

- 'minimalist' cryptography for RFID very active field
 - limited resources takes preference
 - transaction time secondary requirement
- Authentication of shipment
 - RFID used in supply chain (mostly pallet level)
 - Want to make sure items do not get stolen/replaced
 - At item or pallet level?
 - Item authentication ideal but lots of work
 - Authenticating the container does not show what it inside
 - Time to authenticate becomes crucial in supply chain
 - Try to authenticate at item level with pallet level time complexity

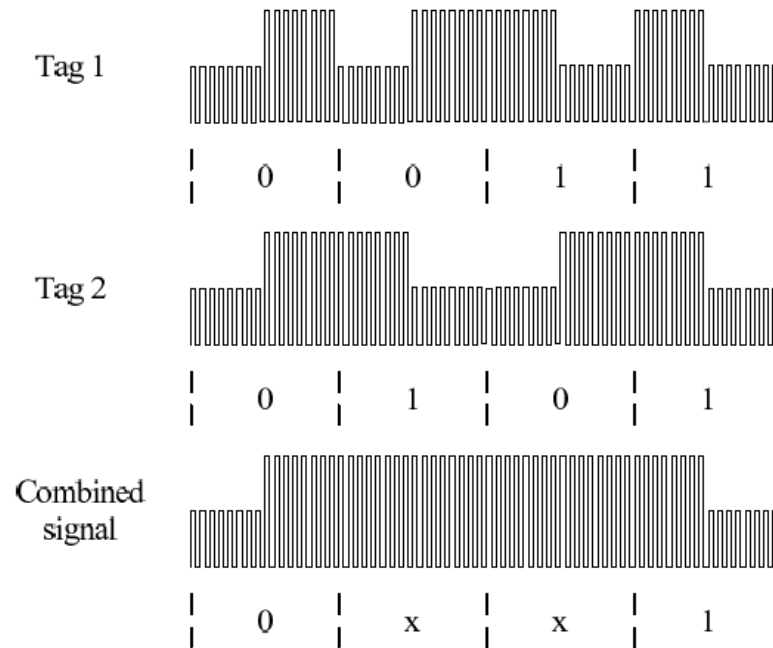
Grouping proofs



■ The ‘yoking’ proof (by Juels)

- Generates proof that two devices are simultaneously present
- Lots of follow up work...
- Not really ‘simultaneous’ – sequential tag interaction

Simultaneous grouping proofs



- Can tags speak at the same time?
 - Yes, results in bit collisions
 - Controlled bit collisions are already used in anti-collision
- Is it possible to use collisions for a simultaneous grouping proof?

Starting off....

β	β'	$\beta \wedge \beta'$
0	0	0
0	1	x
1	0	x
1	1	1

- Some notation
 - Two bit symbols $\beta \beta'$
 - x is a collision
- Collision if any two devices transmit 1 and 0
 - $1 \wedge x = x$ and $0 \wedge x = x$
- Disclaimer: work in progress so might be errors 😊. Please point these out....

Basic security requirement

tags	correct	missing s_4	fake s_4
s_1	01000011	01000011	01000011
s_2	01001001	01001001	01001001
s_3	01100001	01100001	01100001
s_4	11000001	missing	01010001
S	$x1x0x0x1$	$01x0x0x1$	$01xxx0x1$

- Authenticate as a group
 - Group authentication state
- Result should be relevant to the item level
 - The group must be complete (no missing tags)
 - The group must be pure (no fake tags)

Group Initialisation

	Tag states	Choosing bit pairs
s_1	10000000000001000	1, 7 of (1, 2, 3, 4, 5, 6, 7, 8)
s_2	00101000000000000	2, 3 of (2, 3, 4, 5, 6, 8)
s_3	00000010001000000	4, 6 of (4, 5, 6, 8)
s_4	00000000100000010	5, 8 of (5, 8)
S	$x0x0x0x0x0x0x0x0$	

- Only works if each tag contributes collision
 - Cannot leave it to random chance
- Try to construct initial state
 - Choose number of collisions c each of n tags contribute
 - Select c bit pairs for each tag (set first bit to 1)
 - Create group ID and group key and initialise sequence counter
 - Run protocol once to randomise....ship the group off

Permutation function

		<i>Swap</i>	<i>Shift</i>	<i>XOR</i>	
s_1	01000011	s_1	01000011	11010000	01110101
s_2	01001001	s_2	01001010	10010010	00110111
s_3	01100001	s_3	01010010	10010100	00110001
s_4	11000001	s_4	11000010	10110000	00010101
S_{old}	$x1x0x0x1$	S_{new}	$x10xx01x$	$1xx10xx0$	$0xx10xx1$
			$\underbrace{0101}_{\text{Swap}}$	$\underbrace{01}_{\text{Rotate}}$	$\underbrace{10100101}_{\text{XOR}}$

- Update to group authentication state

- Should preserve the information in our constructed state

$$f_2(s_1) \wedge f_2(s_2) \wedge \dots \wedge f_2(s_n) = f_2(S)$$

- Permutation should not be predictable

- Permutation built on

- Swap, rotate and XOR based on result of keyed pseudo-random function
- Each operation preserves state information

Protocol Assumptions: System Operation

- The system consists of a number of nodes that track the progress of a group of items, i.e. a single package or shipment, from its sender to the intended recipient.
- The sender, recipient and nodes could be controlled by different organisations, but a key management infrastructure is in place that allows the sender to distribute key material to the verifying nodes and the recipient, i.e. the sender can securely share information with the verifying nodes and the recipient.
- The group verifier, i.e. a node or the recipient, does not necessarily have the ability to share information with other verifiers and as a result it should not require knowledge of previous protocol runs between the group and other verifiers.

Protocol Assumptions: Security Objectives

- The purpose of our protocol is only to prove the completeness and purity of a chosen group to the recipient and intermediate verifying nodes..
- The protocol does not provide non-repudiation of purity and completeness for anyone who does not trust the verifying node or recipient.
- The sender, recipient and verifying nodes are seen as trusted entities.
- Privacy – attacker cannot gain information about individual tags although group can be tracked. Prior to tags possessed by end user privacy concerns limited.

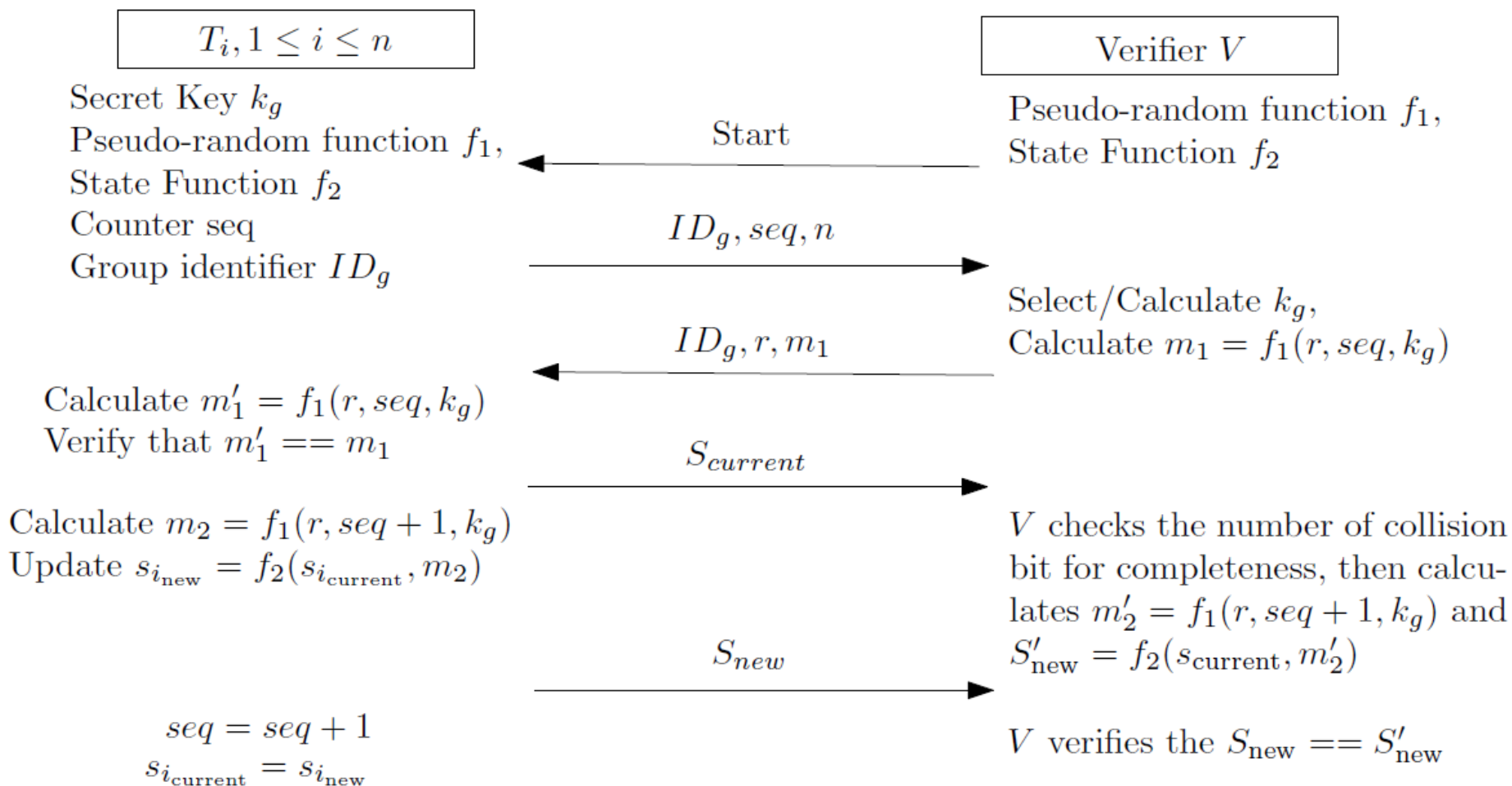
Protocol Assumptions: Cryptographic Primitives

- A group of tags, sender, recipient and the verifying nodes share a dedicated secret group key k_g , a keyed public pseudo-random function f_1 and a public bit permutation function f_2 .
- Even though not all RFID devices are currently tamper resistant, we assume that tags used in this scheme can reliably store a secret key and that the sender, recipient and verifying nodes can calculate this key at runtime.

Protocol Assumptions: Grouping Process

- A group would be a number of items physically packaged together as these must be interrogated together by a single reader.
- The sender is responsible for creating a legitimate group and initialising the tags.
- The sender will send the group ID and a description of the items to the intended recipient to enable the recipient to identify the contents.
- If required, tags can contain additional data protected with a key shared between the sender and recipient, although this is beyond the scope of our protocol.

Protocol



Security analysis

- Simple security issues
 - Privacy
 - Group keys
 - Tag desynchronization
- Relay....?
 - Whole group – yes
 - Subset of tags – practical resistance (attacker's relay time limited)
- Look at some special cases
 - Theoretical attack probability
 - Associated experiments
 - Practical results close to theoretical probability

Security Analysis: Case 1

- The attacker replaces tags with 'normal' tag
 - Adheres to protocol rules
 - Not aware of what other tags are sending
- Guess the tag state that will result in correct group state
 - In bit positions where other tags contribute collisions the attacker need not guess correctly (any error masked by collisions).
 - Attack success probability if attacker removes n_a tags:

$$p_a = \left(\frac{1}{2}\right)^{c(n+n_a)}$$

Security Analysis: Case 2

- The attacker replaces tags with 'quiet' tag
 - The tag does not adhere to protocol rules
 - Guess only where it should contribute collisions (otherwise stay quiet)
 - Not aware of what other tags are sending
- Guess correct bit collision position and bit value causing collision
 - The tag does not need to guess the values of non-collision bit positions
 - Attack success probability if attacker removes n_a tags:

$$p_a = \binom{cn}{cn_a}^{-1} \cdot \left(\frac{1}{2}\right)^{2cn_a}$$

Security Analysis: Case 3

- The attacker replaces tags with 'smart' tag
 - The tag does not adhere to protocol rules
 - The tag is aware of what other tags are sending
- Guess correct bit collision position
 - Tags knows where other tags cause collisions
 - If no collision tag must decide to cause collision
 - It knows what bit to transmit to cause the collision
 - Attack success probability if attacker removes n_a tags:

$$p_a = \binom{(cn + cn_a)/2}{cn_a}^{-1} \cdot \left(\frac{1}{2}\right)^{cn_a}$$

Security Analysis: Case 4

- Attacker knows tag state and collision positions
 - Worst case
 - Not very practical to reach this position
- Guess correct bit collision position
 - Tag knows the bit pairs it has to contribute collisions
 - Smart tags: has to guess if the bit pair swops and how much pairs rotate
 - Quiet tags: also has to guess bit value that will cause collision
 - Attack success probability if attacker removes n_a tags:

Smart tags

$$p_a = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{cn_a}$$

Quiet tags

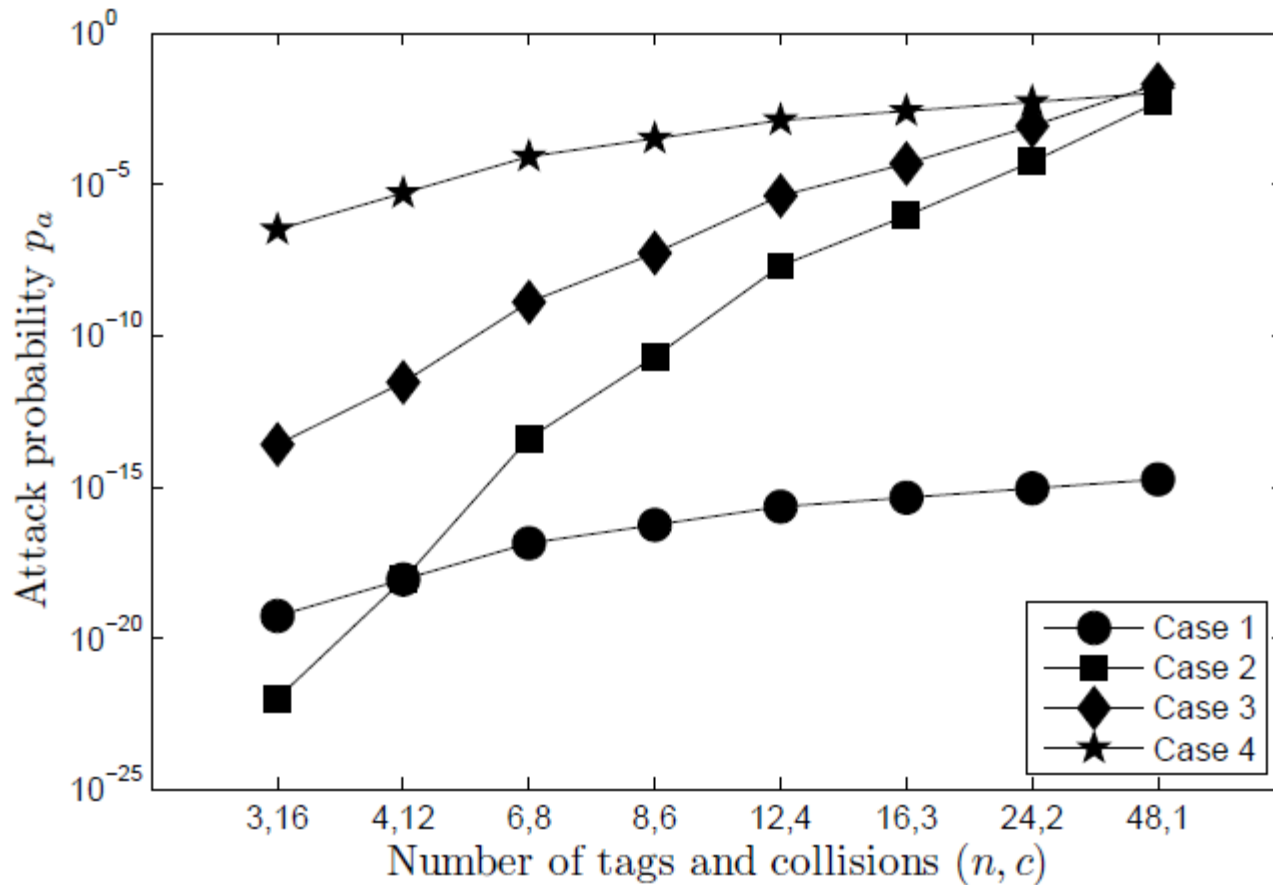
$$p_a = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{2cn_a}$$

Security Analysis: Case 5

- The attacker creates a new group
 - Takes all the tags
 - Tries to guess entire group state for next protocol run

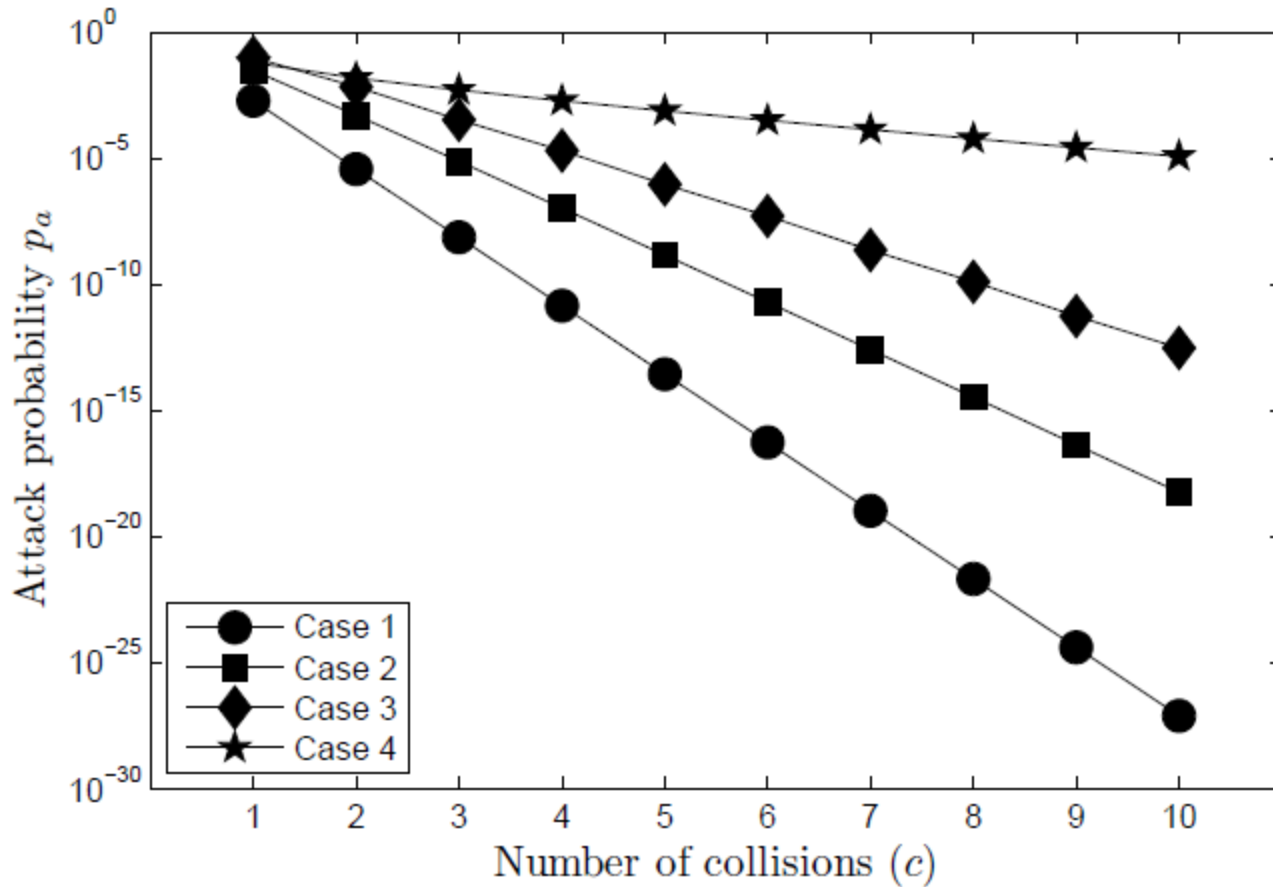
$$p_a = \left(\frac{1}{2}\right)^{2cn}$$

Result 1



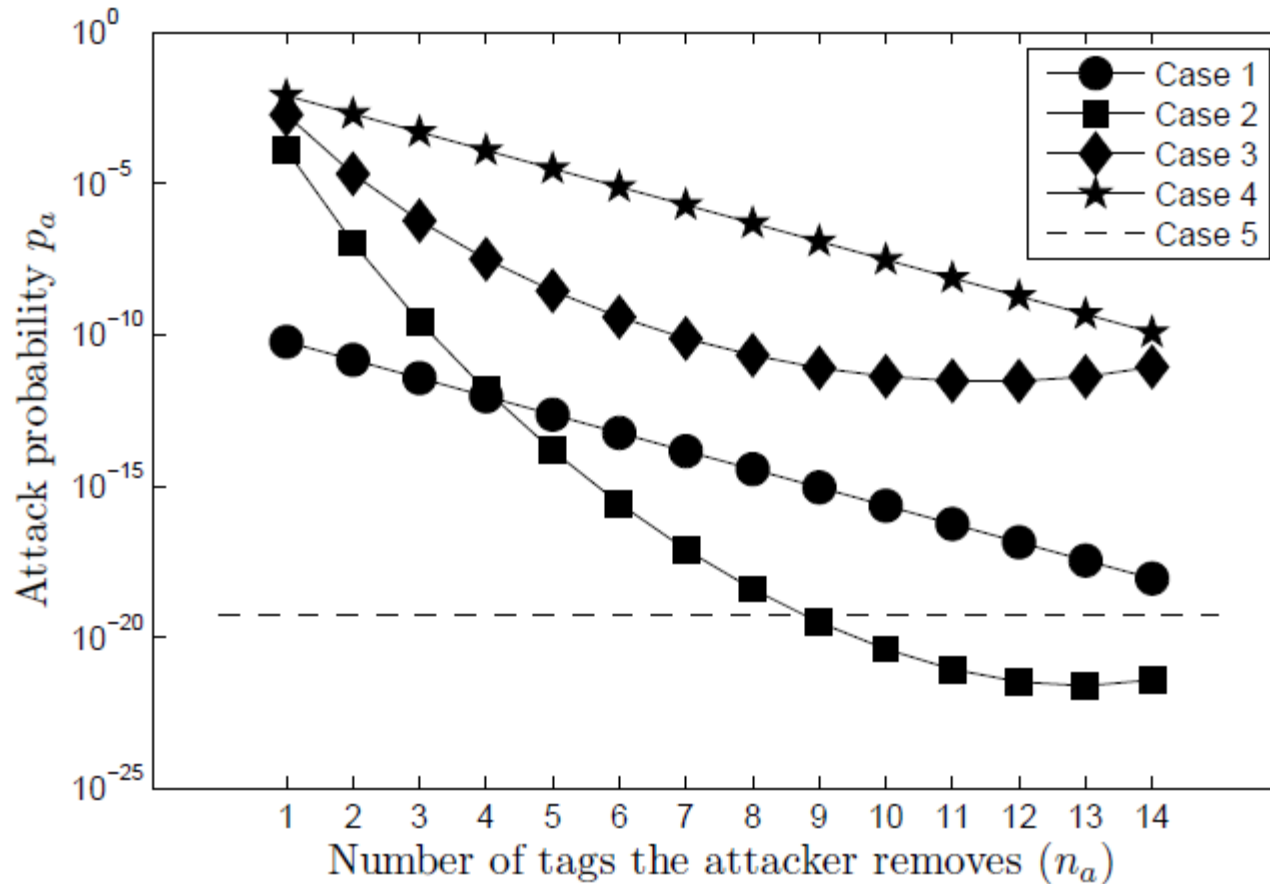
- Fixed state size: 96 bits (response length for ISO 15693/18000-6)
- Attacker takes one tag

Result 2



- Group size ($n=8$), increased number of collisions (state size)
- Attacker takes one tag

Results 3



- Attacker takes multiple tags (group size 16, 2 collisions per tag)

Practical considerations

- **RFID system architecture**
 - Single antenna system needed
 - These are widely used
- **Practical Group Size**
 - Reader technology limits number of tags simultaneously readable
 - Size of group determines state length
 - Issues with large groups, but beneficial to multiple small/medium groups
 - Example: Medicine – blister packs
- **RFID technology in supply chains**
 - This scheme can be implemented with current physical layer standards
 - Tags can synchronize – they do so already for anti-collision

Thank you!

Any questions?

Get in touch

gerhard.hancke@rhul.ac.uk