



UNIVERSIDAD  
DE MÁLAGA



Ingeniería de Comunicaciones

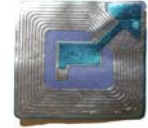
# A Flyweight RFID Authentication Protocol

Mike Burmester & Jorge Munilla

*Workshop on RFID Security 2009*

**RFIDSec 09 Leuven**

E.T.S.Ingeniería de Telecomunicación  
Campus de Teatinos, 29071 Málaga



# Index

## 1.- Introduction:

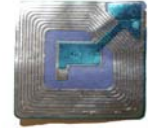
- 1.a EPCGen2 Standard
- 1.b CRC
- 1.c RNG

## 2.- Analysis of some recently proposed protocols

- 2.a Chen-Deng (2009)
- 2.b Sun-Ting (2009)
- 2.c Quingling-Yiju-Yonghua (2008)
- 2.d Seo-Baek (2009)
- 2.e Choi-Lim (2008)

## 3.- A Flyweight RFID Authentication Protocol

- 3.a Requirements
- 3.b Protocol Description
- 3.c Refreshing
- 3.d Security analysis



## 1.a EPCGen2 Standard

- ✓ UHF 860-960 MHz.
- ✓ Two layers: Physical and Tag-Identification
- ✓ Tag population management: Select, Inventory and Access

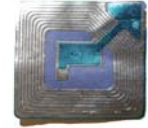
*R→T Query with the parameter  $Q$  (and QueryRep)*

*T→R RN16*

*R→T ACK(RN16)*

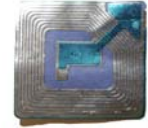
*T→R EPC Data (EPC, PC)*

Inventory protocol



## 1.a EPCGen2 Standard

- ✓ UHF 860-960 MHz.
- ✓ Two layers: Physical and Tag-Identification
- ✓ Tag population management: Select, Inventory and Access
- ✓ Link-layer coding and two 32-bit passwords (Kill and Access)
- ✓ Hardware requirements (reduced power consumption) :
  - CRC
  - RNG



## 1.b CRC

### ✓ CRC( Cyclic Redundancy Code)

- $B(x) \cdot x^n = d(x) \cdot g(x) + r(x) \rightarrow r(x) = \mathit{CRC}(B) = (B(x) \cdot x^n) \bmod g(x)$

- Properties:

$$\mathit{CRC}(A \oplus B) = \mathit{CRC}(A) \oplus \mathit{CRC}(B)$$

$$\mathit{CRC}(A \cdot x^{n \cdot (k-1)}) = \mathit{CRC}(\mathit{CRC}(\dots \mathit{CRC}(A))) \rightarrow \mathit{CRC}^k(A)$$

### ✓ EPCGen2

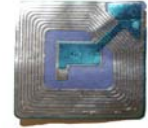
- CRC-CCITT  $g(x) = x^{16} + x^{12} + x^5 + 1$

- Initialized with ones  $\mathit{CRC}(B) = (B(x) \cdot x^{16}) \bmod g(x) + \mathit{CRC}(0)$

- Properties:

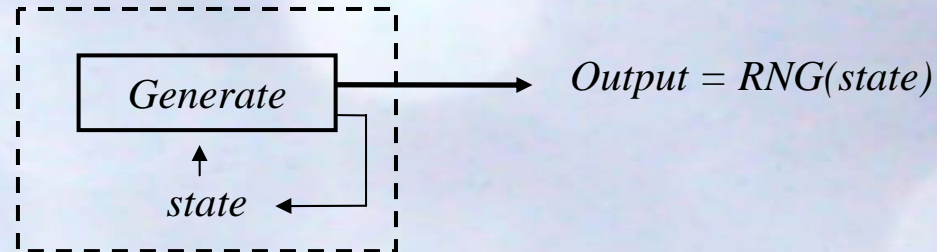
$$\mathit{CRC}(A \oplus B) = \mathit{CRC}(A) \oplus \mathit{CRC}(B) \oplus \mathit{CRC}(0)$$

$$\mathit{CRC}(A \cdot x^{16 \cdot (k-1)}) = \mathit{CRC}^k(B) \oplus \mathit{CRC}^{k-1}(0)$$



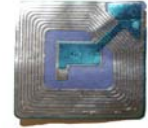
## 1.c RNG

### ✓ RNG (Pseudo Random Number Generator)

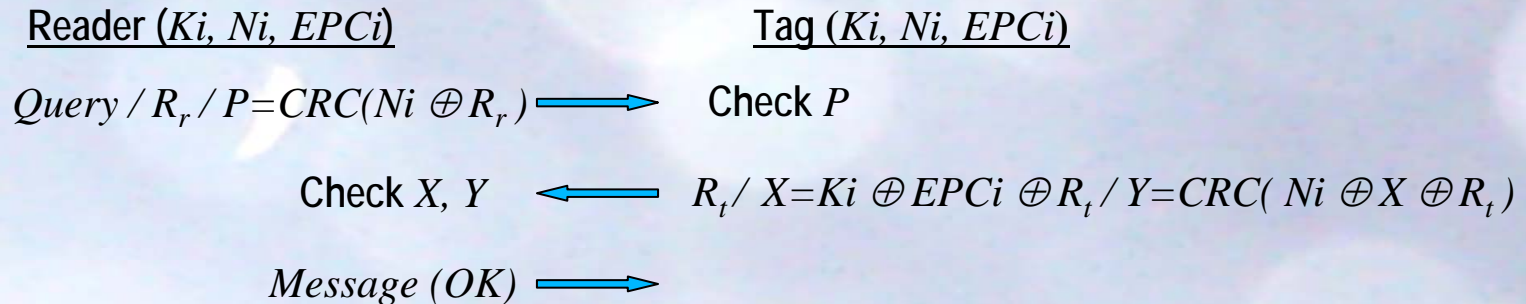


### ✓ EPCGen2

- Probability of a single RN16  $0.8/2^{16} < P(RN16=j) < 1.25/2^{16}$
- Collisions for a tag population of 10.000, Prob < 0.1%
- Predicting an RN16 Prob < 0.25% if the prior draws are known

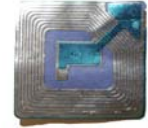


## 2.a Analysis of Chen-Deng Protocol

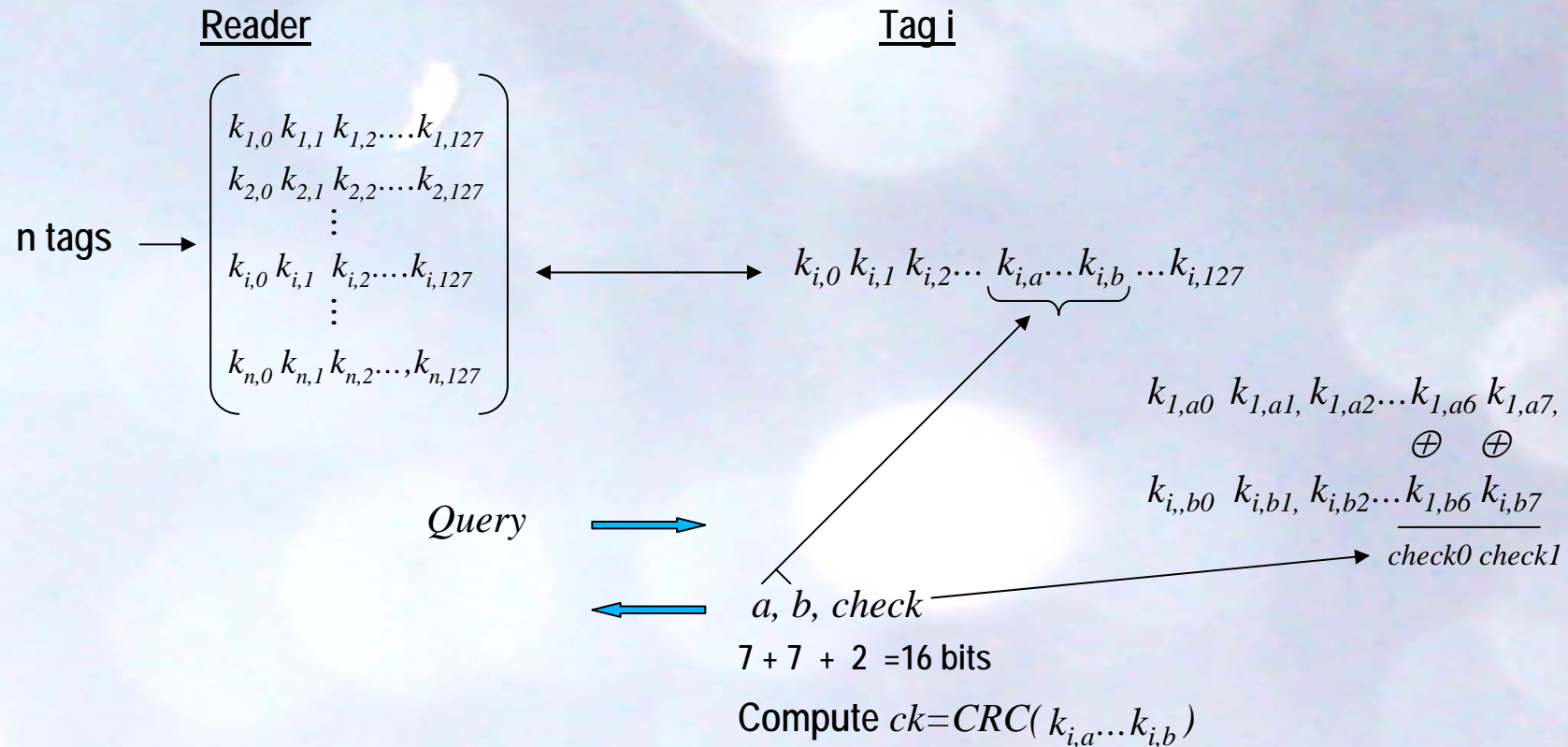


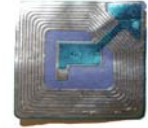
### Weaknesses:

- Tag impersonation. Replay attack. Tag's response does not depend on  $R_r$
- New valid  $R_t^* / X^* / Y^*$  can be computed if a previous authentication has been eavesdropped
  - $R_t^* = R_a$
  - $X^* = X \oplus (R_t \oplus R_a)$
  - $Y^* = Y$  (does not change because  $Y = CRC(N_i \oplus K_i \oplus EPC_i)$  is constant for the  $i$ -tag)
- Reader can be also impersonated by replaying  $P$  or computing new valid  $P^*$ .
  - $P^* = P \oplus CRC(R_r \oplus R_a) \oplus CRC(0)$

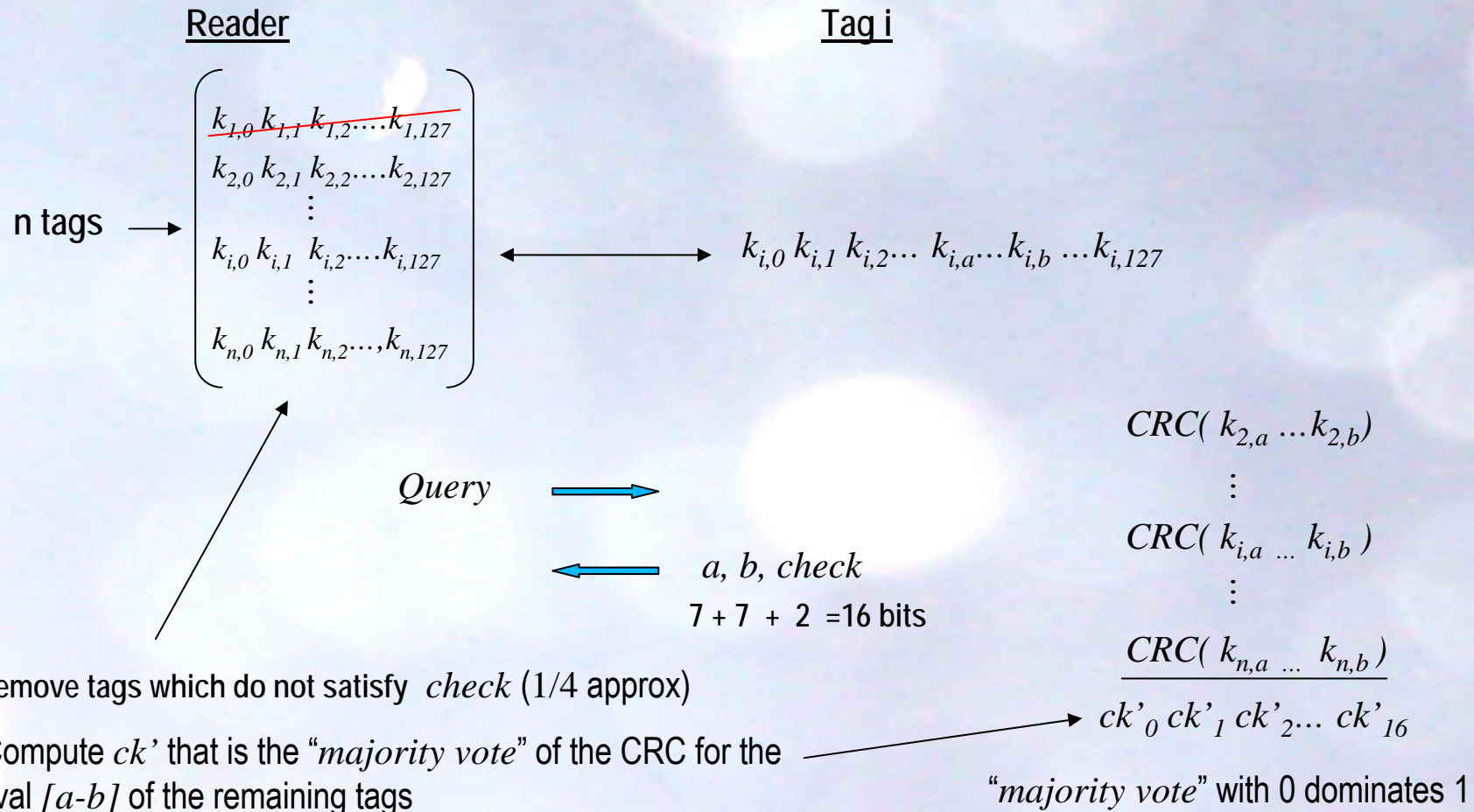


## 2.b Analysis of Gen2+ (Sung-Ting)

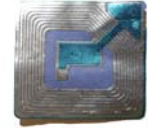




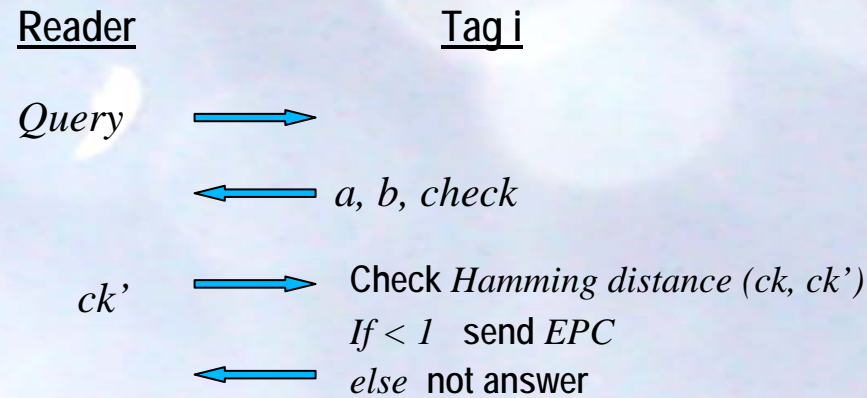
## 2.b Analysis of Gen2+ (Sung-Ting)







## 2.b Analysis of Gen2+ (Sung-Ting)

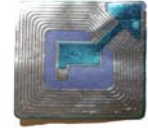


### ✓ Weaknesses:

- Tag impersonation. Replay attack. The tag chooses the intervals:  $a, b$
- It is subject to a more complicated analysis attack to know the key material. The adversary can ask for specific words ( $a, b, check = a, a, 00$ ) and the reader provides him with information about the CRC of those words.

To impersonate the reader, the CRC of any interval can be computed from the CRC of its words.

$$CRC(A // B) = CRC(A \oplus B \cdot x^{16}) = CRC(A) \oplus CRC(B \cdot x^{16}) \oplus CRC(0)$$



## 2.c Analysis of Quingling-Yiju-Yonghua protocol

Reader ( $aPW$  (32bits),  $TIDl$ ,  $TIDh$ )

Tag ( $aPW$  (32bits),  $TIDl$ ,  $TIDh$ )

Query /  $R_r$   $\longrightarrow$

Check  $M$   $\longleftarrow$   $R_t$  /  $M = (Ml//Mh) \oplus aPW$

$$Nl = CRC(TIDl \oplus R_r)$$

$$Nh = CRC(TIDh \oplus R_r)$$

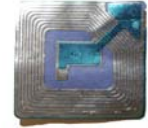
$N = (Nl//Nh) \oplus aPW$   $\longrightarrow$  Check  $N$

$$Ml = CRC(TIDl \oplus R_r \oplus R_t)$$

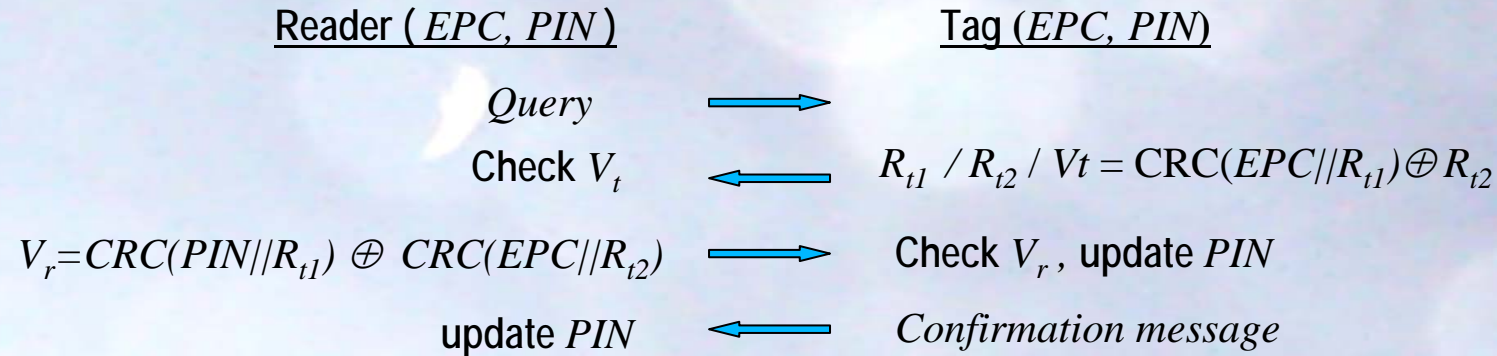
$$Mh = CRC(TIDh \oplus R_r \oplus R_t)$$

### ✓ Weaknesses:

- Tag impersonation: new valid  $R_t^*$  and  $M^*$  for a new  $R_r^*$  can be computed by using DATA from an eavesdropped interrogation
  - $R_t^* = Ra$
  - $M^* = M \oplus (A//A)$  where  $A = CRC(R_r \oplus R_r^* \oplus R_t \oplus Ra) \oplus CRC(0)$
- Reader impersonation (previous eavesdropped interrogation is not needed)
  - $R_r^* = Ra$  (tag answers with  $M^*$ )
  - $N^* = M^* \oplus (B//B)$  where  $B = CRC(R_r^*) \oplus CRC(0)$

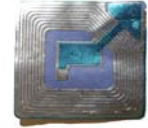


## 2.d Analysis of Seo-Baek protocol- scheme 1

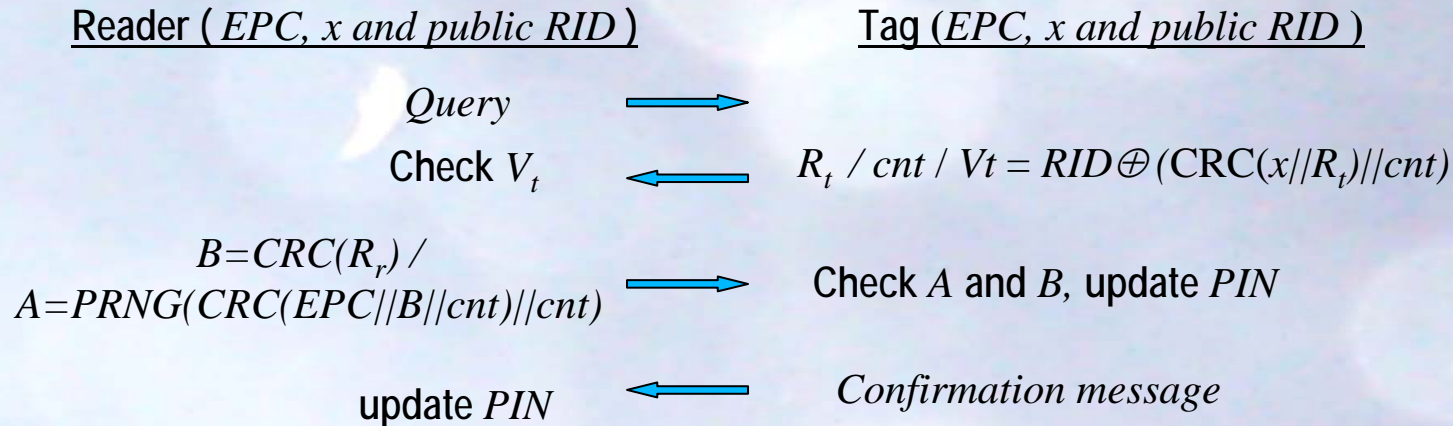


### ✓ Weaknesses:

- Tag impersonation. Replay attack;  $V_t$  can be repeated.
- *EPC* and *PIN* can be disclosed as previously explained because the CRC properties.
- Synchronization problems

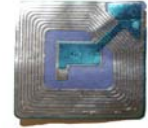


## 2.d Analysis of Seo-Baek protocol- scheme 2

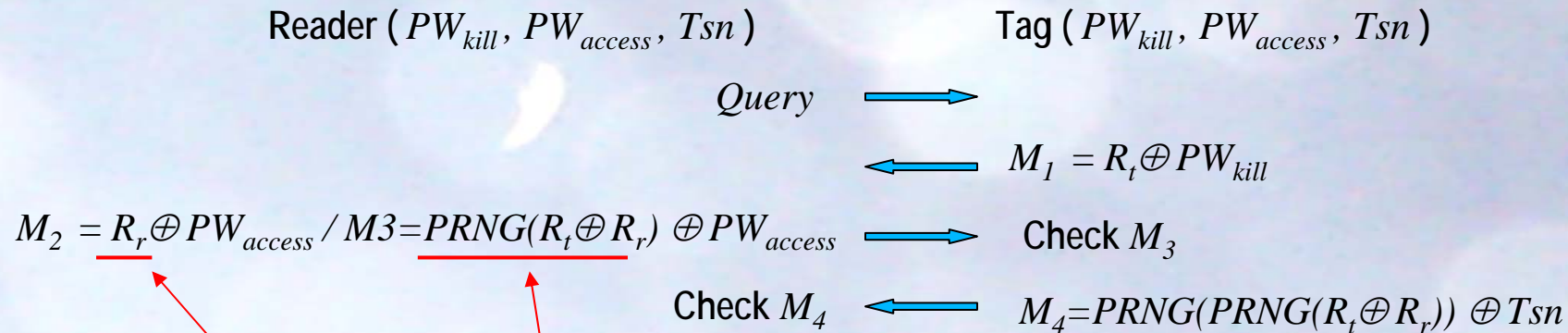


### ✓ Weaknesses:

- Tag impersonation. Replay attack. An adversary can impersonate the reader and get valid  $R_t / cnt / V_t$
- $x$  can be disclosed as previously explained because the CRC properties.
- Synchronization problems



## 2.e Analysis of Choi-Lim protocol

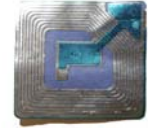


### ✓ Weaknesses:

- The reader can be impersonated. New valid  $M_2^*$  and  $M_3^*$  can be computed for a new  $M_1^*$  by using DATA from an eavesdropped interrogation
  - $M_3^* = M_3$
  - $M_2^* = M_2 \oplus M_1^* \oplus M_1 \rightarrow (R_r^* = R_r \oplus R_t^* \oplus R_t)$



It could be subject to related key attacks.



# Index

## 1.- Introduction:

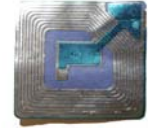
- 1.a EPCGen2 Standard
- 1.b CRC
- 1.c RNG

## 2.- Analysis of some recently proposed protocols

- 2.a Chen-Deng (2009)
- 2.b Sun-Ting (2009)
- 2.c Quingling-Yiju-Yonghua (2008)
- 2.d Seo-Baek (2009)
- 2.e Choi-Lim (2008)

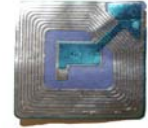
## 3.- A Flyweight RFID Authentication Protocol

- 3.a Requirements
- 3.b Protocol
- 3.c Refreshing
- 3.d Overview



## 3.a Protocol's Requirements

- ✓ Authentication → Mutual authentication
  - ✓ Privacy (traceability) → Session unlinkability  
(privacy between successful interrogations)
- +
- ✓ Maximum communication efficiency → 3 flows (optimistic)
  - ✓ Minimum computational complexity → "just" a RNG
- } Flyweight
- +
- ✓ Forward security (previous interrogations cannot be linked even if the tag is compromised)
  - ✓ (Weak) Backward security



### 3.b Description-Optimistic case

Reader ( $RN_1^{cur}, RN_2, RN_3, RN_4, RN_5, RN_1^{next}, ID_{tag}, g_{tag}, K^r$ )

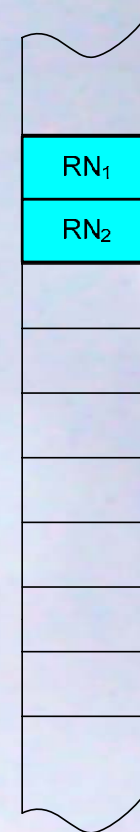
Tag ( $RN_1, RN_2, ID_{tag}, g_{tag}, K^r$ )

Query →

$RN_1^{cur}$	ID	$RN_2$ $RN_3$ $RN_4$ $RN_5$ $RN_1^{next}$	RNG(g)
2345	x1	y11 y12 y13 y14 y15	z1
		⋮	
12175	xi	yi1 yi2 yi3 yi4 yi5	zi
23450	$ID_{tag}$	$RN_2$ $RN_3$ $RN_4$ $RN_5$ $RN_1^{next}$	$g_{tag}$
		⋮	
62175	xn	yn1 yn2 yn3 yn4 yn5	zn

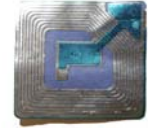
Look up in DB ←  $RN_1$  (e.g. 23450)

RNG( $g_{tag}$ )

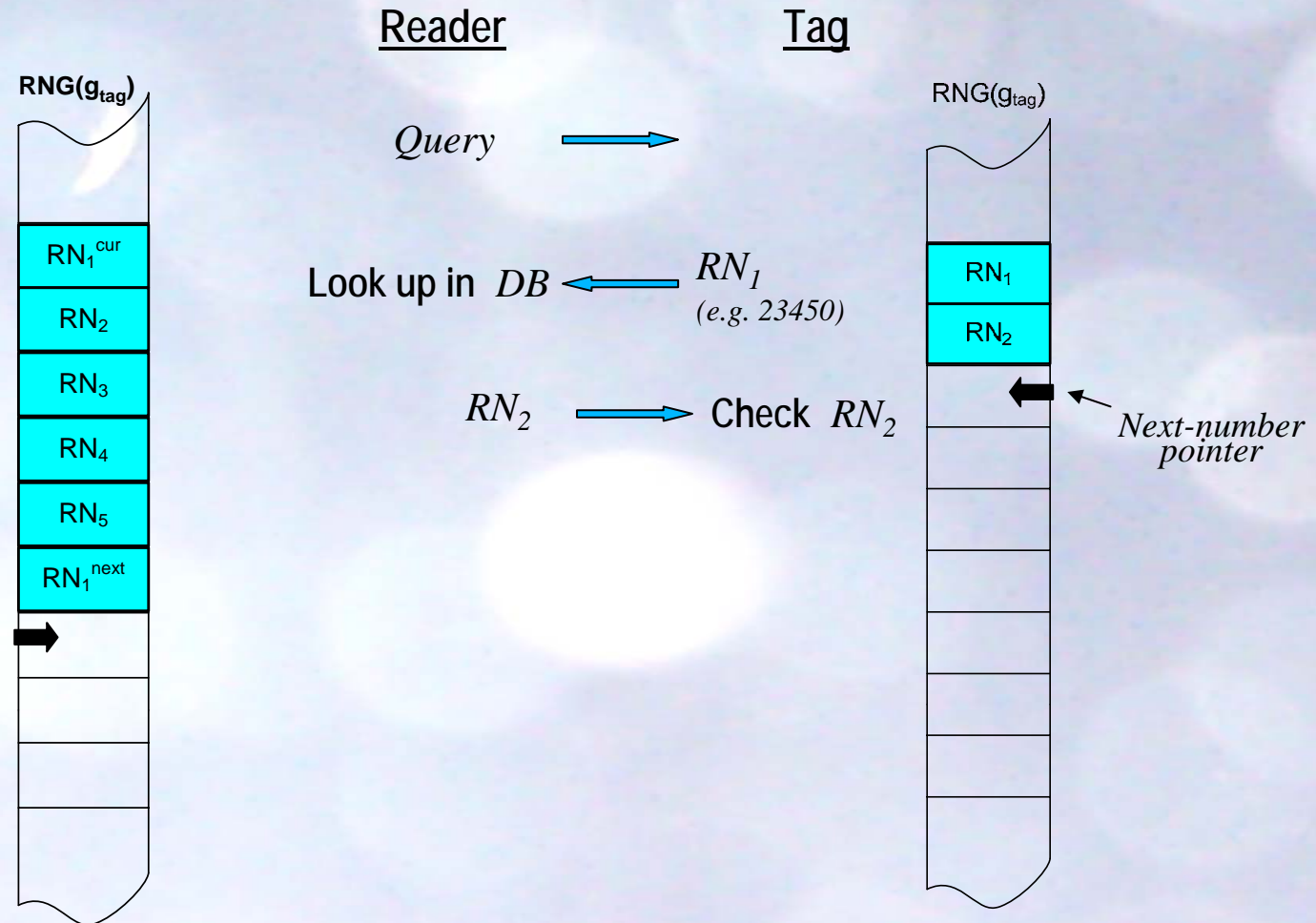


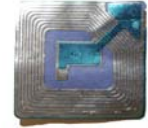
Non-volatile memory

If  $RN_1$  is not found, try with the table indexed by  $RN_1^{next}$

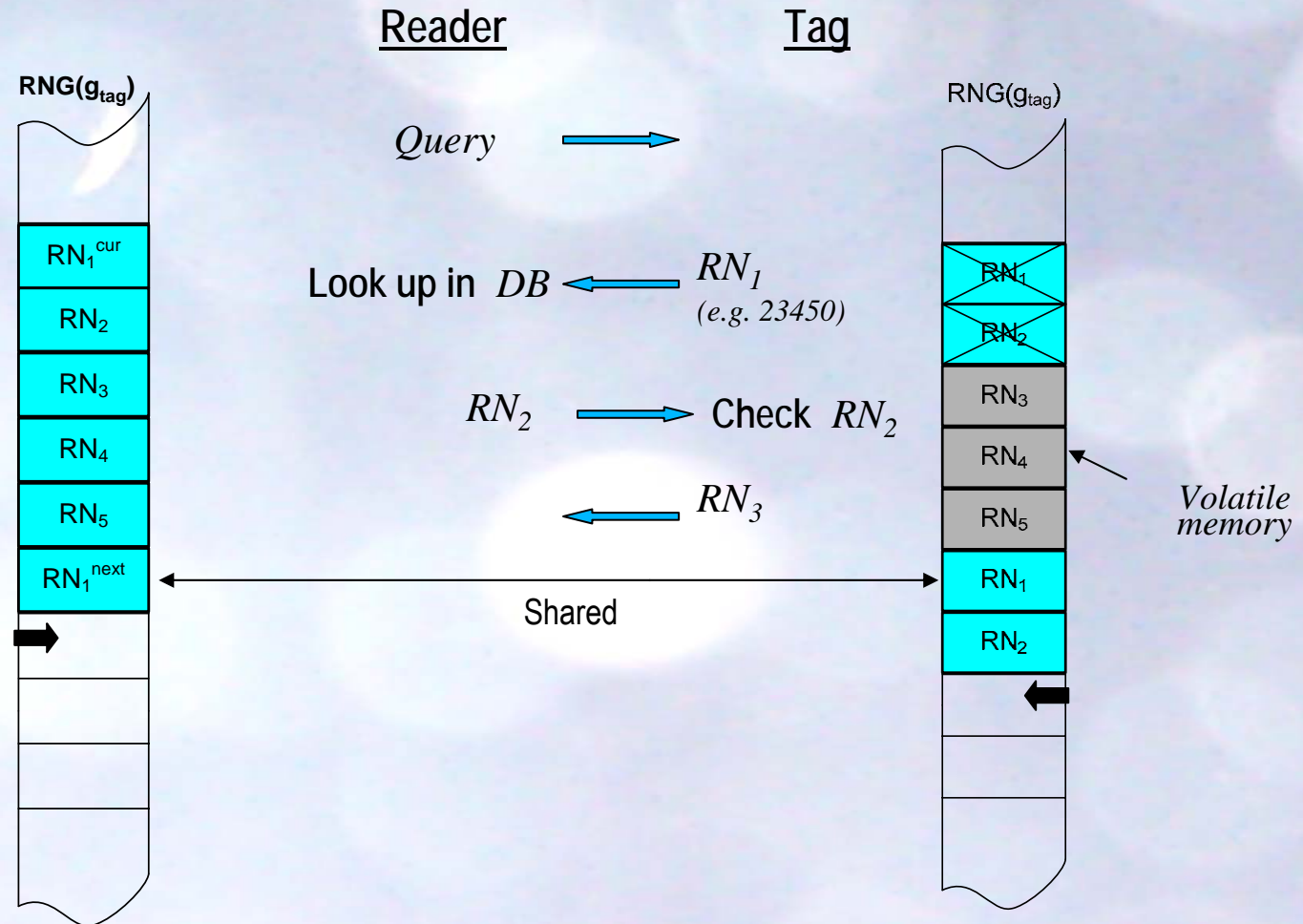


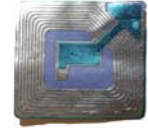
### 3.b Description-Optimistic case



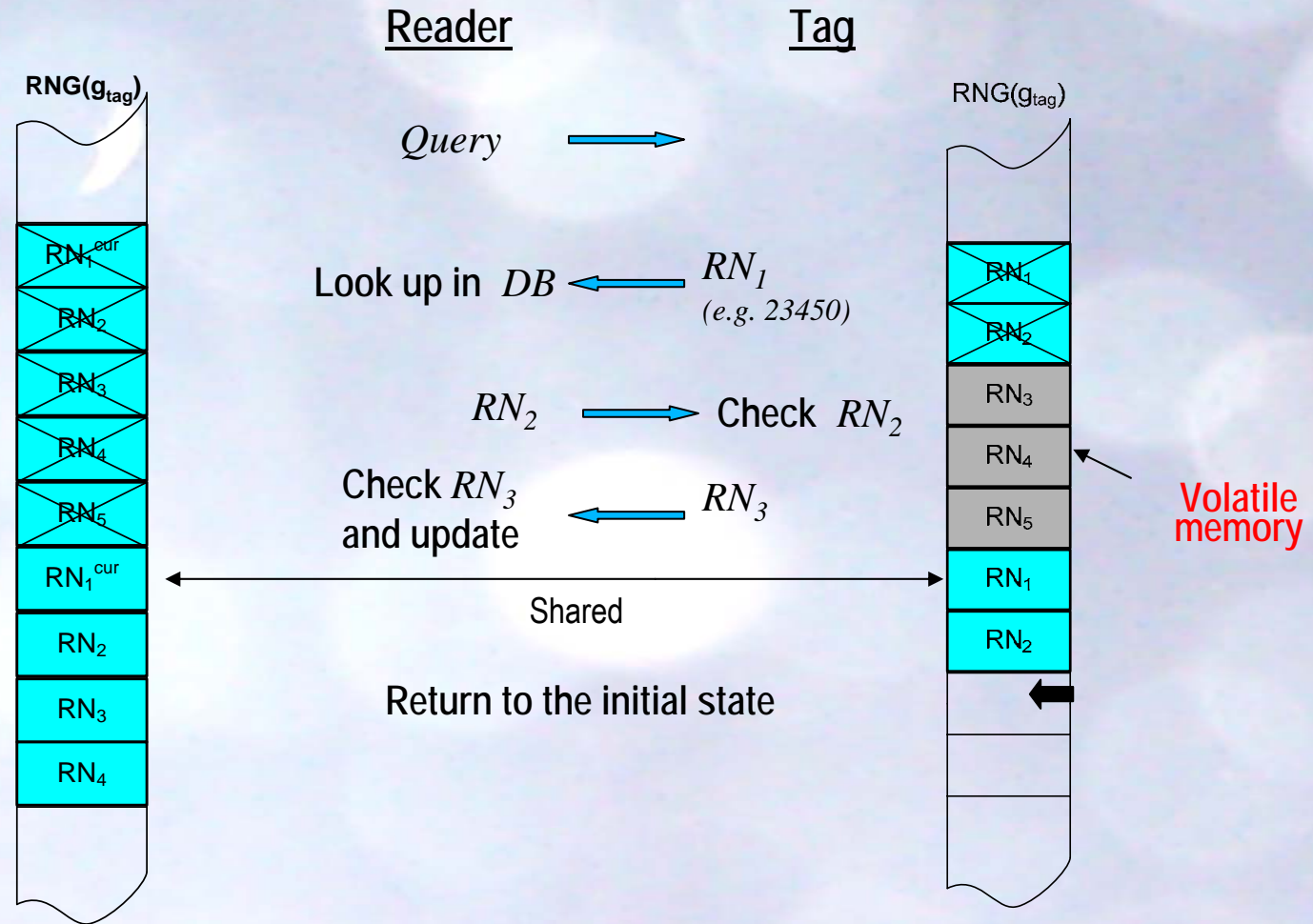


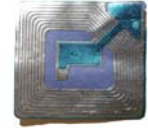
### 3.b Description- Optimistic case



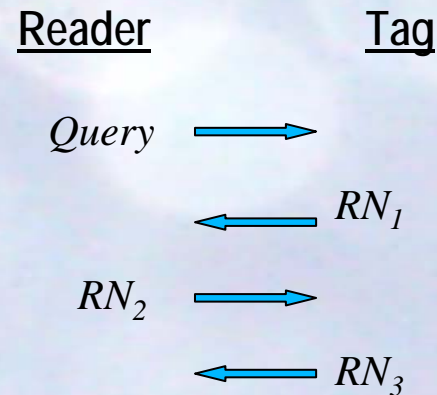


### 3.b Description- Optimistic case



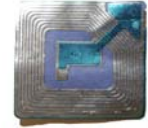


## 3.b Description- Optimistic case



Similar to the original EPCGen2 (it can be easily embedded)

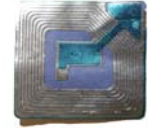
R→T *Query with the parameter Q (and QueryRep)*  
T→R *RN16*  
R→T *ACK(RN16)*  
T→R *EPC Data (EPC, PC)*



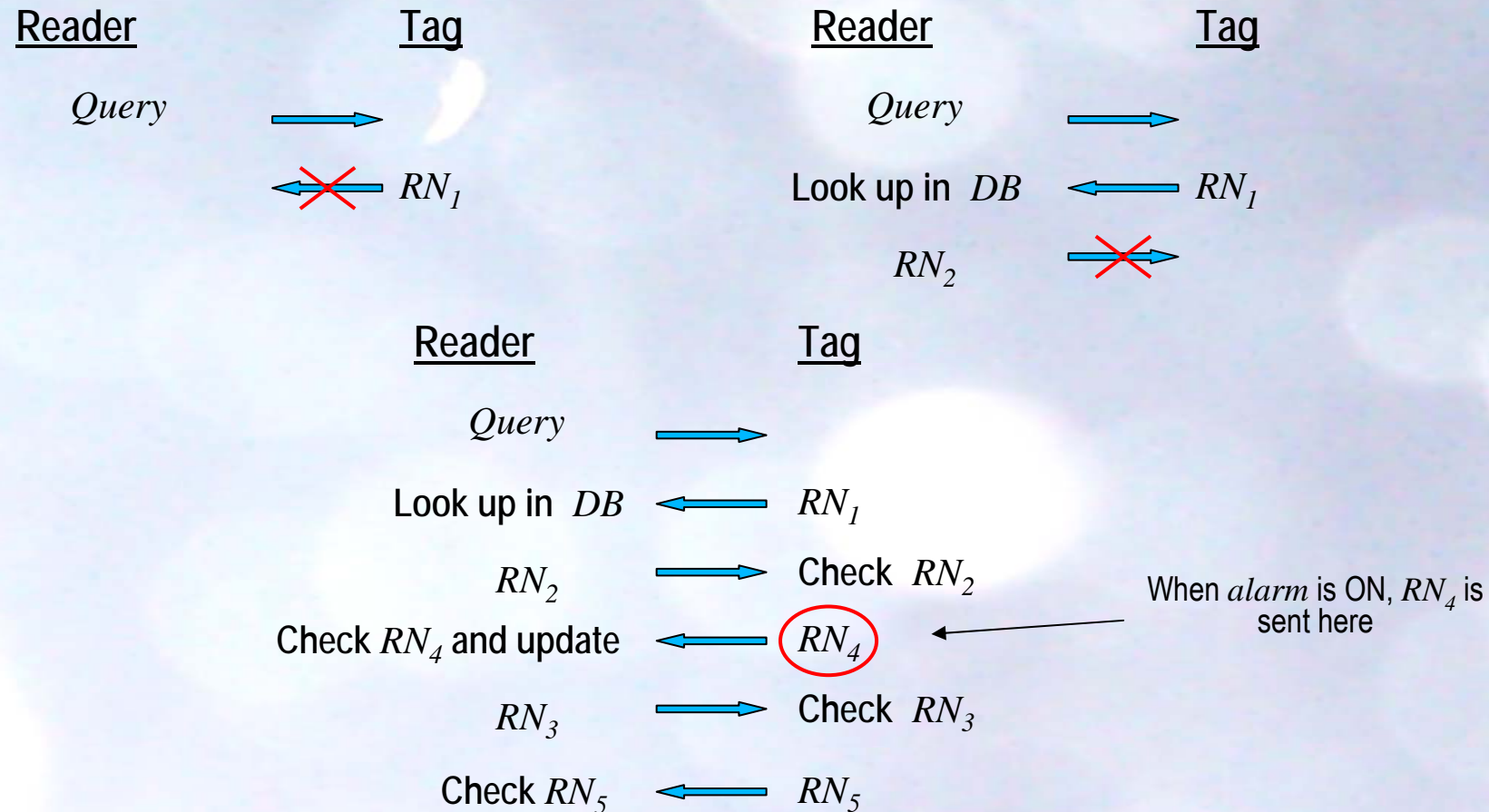
### 3.b Description- Incomplete Interrogations-Case I

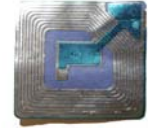


- ✓ alarm (*alarm* and *alarm'*) = ON. The protocol changes and two more flows are required

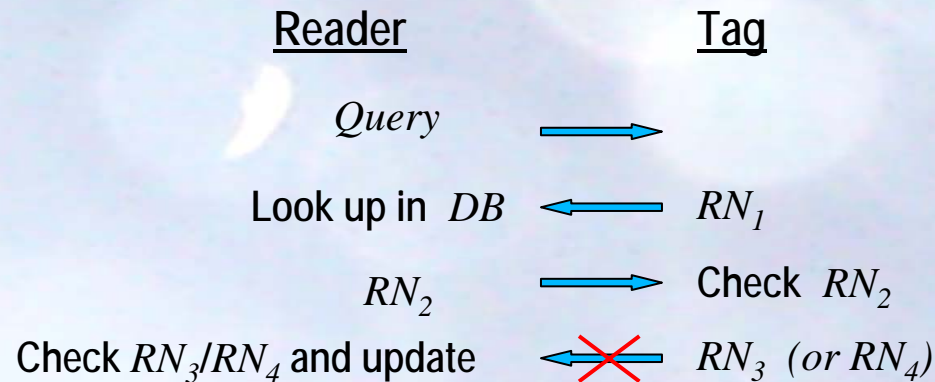


### 3.b Description- Incomplete Interrogations-Case I

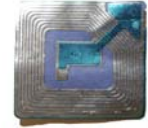




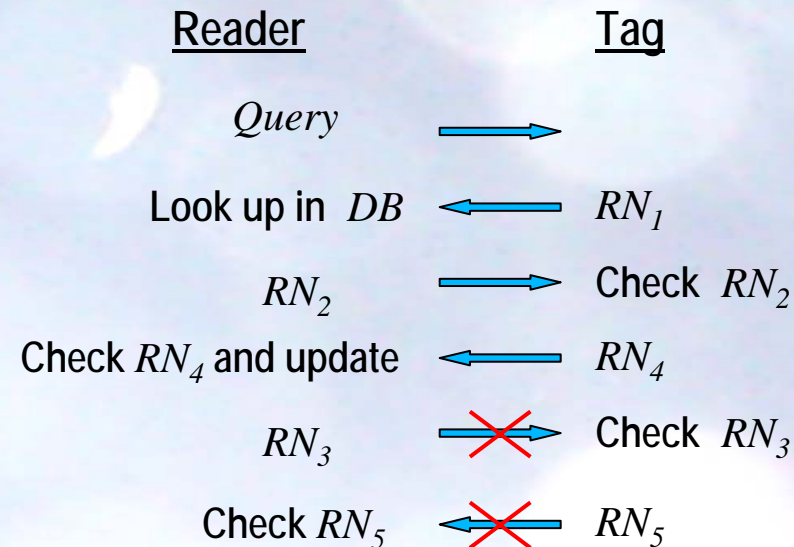
### 3.b Description- Incomplete Interrogations- Case II



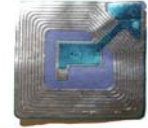
- ✓ Tag has updated, Reader has not.  $RN_1^{next} \longleftrightarrow RN_1$
- ✓ Tag will not use current  $RN_3$ ,  $RN_4$  and  $RN_5$  anymore (volatile memory).
- ✓ Adversary cannot get  $RN_5$  to complete the protocol.
- ✓ Reader does not wait for  $RN_3/RN_4$  indefinitely : timers are used to close the sessions.



### 3.b Description- Incomplete Interrogations- Case III

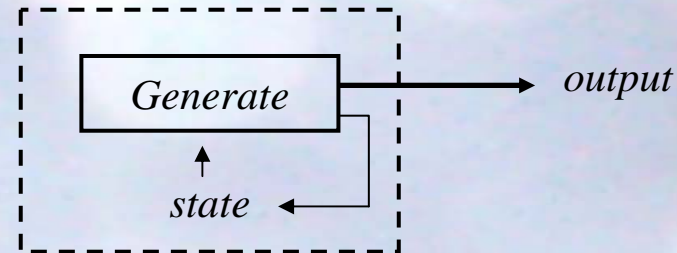


- ✓ Both parties have updated.  $RN_1^{cur} \longleftrightarrow RN_1$
- ✓  $RN_3$ ,  $RN_4$  and  $RN_5$  will be not used (accepted) anymore .
- ✓ Reader does not wait for  $RN_5$  indefinitely : timers are used to close the sessions.

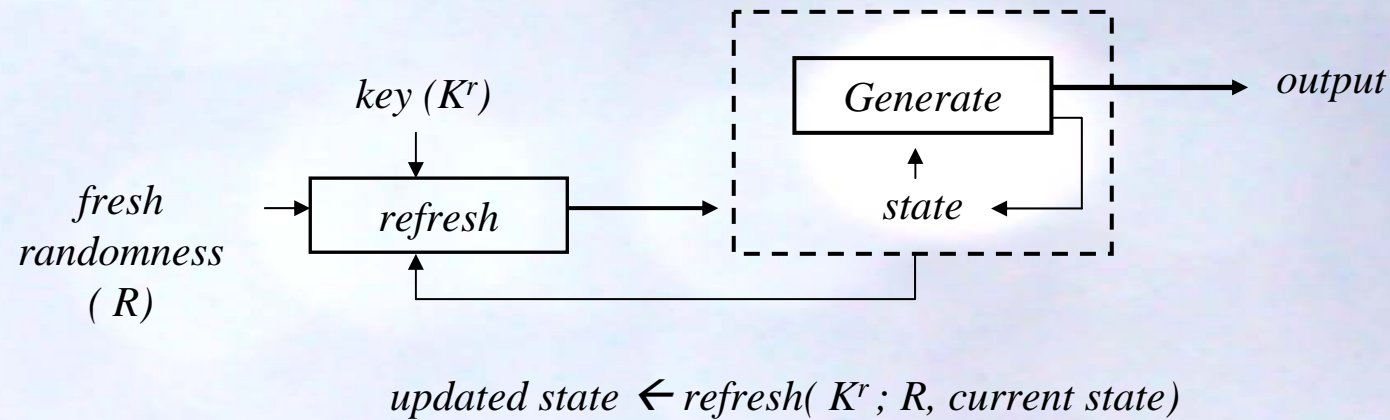


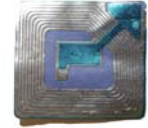
## 3.c Refreshing

✓ RNG



✓ RNG with refreshing





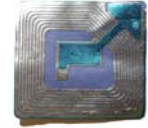
## 3.c Refreshing

- ✓ Objectives:
  - Make harder to determine the state
  - Restrict the impact if the state is eventually determined (“weak backward security”).
- ✓ Refreshing would not be necessary if RNG was good enough (most protocols assume it)
- ✓ When refresh? It will depend on the used RNG.
- ✓ “Weak” Backward Security: an adversary cannot be authenticated after refreshing even if the current state of the RNG is compromised.

$updated\ state \leftarrow refresh(K^r; R, current\ state)$

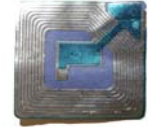
Backward

Forward



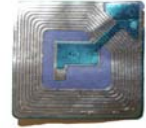
## 3.c Refreshing

- ✓ Objectives:
  - Make harder to determine the state
  - Restrict the impact if the state is eventually determined (“weak backward security”).
- ✓ Refreshing would not be necessary if RNG’s were good enough (most protocols assume it)
- ✓ When refresh? It will depend on the used RNG.
- ✓ Backward Security: an adversary cannot be authenticated after refreshing even if the current state of the RNG is compromised.
- ✓ Example of implementation:  $refresh(K_r; R, current\ state) = RNG(K_r \oplus R \oplus current\ state)$



## 3.d Overview

- ✓ **Availability** → server and tag are always synchronized (sharing a number)
- ✓ **Mutual authentication and session unlinkability** → if the output of the RNG cannot be predicted (look random).
- ✓ **Forward Security** → if RNG is a one-way function or refreshing is used
- ✓ **Backward Security** → with refreshing
- ✓ **Timers are used after sending a message to close the session after a certain time.**  
The protocol is subject to MIM relay attacks that relay messages faster than this.
- ✓ **Optimistic:** just 3 flows are required (5 in the non-optimistic case)
- ✓ **Easily embeddable in EPCGen2**
- ✓ **Not resource demanding:** just a synchronized RNG



*Thank you for your attention*

*Any Question?*



*A Flyweight RFID Protocol*