A MP-LFSR based Pseudorandom Number Generator for EPC Gen2 Systems

Mitacs Workshop on Network Security & Cryptography

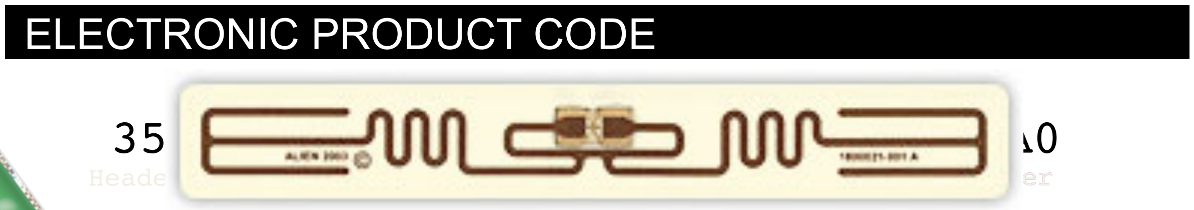
Joaquin Garcia-Alfaro
Institut TELECOM & UOC/IN3

Joint work with
J. Herrera-Joancomarti and J. Melia-Segui

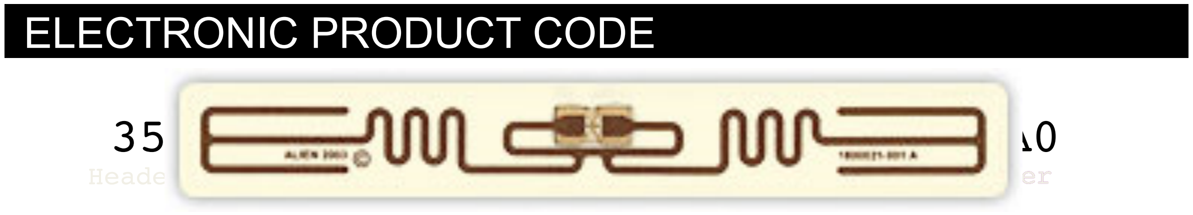
EPC: Electronic Product Code

- Family of coding schemes to uniquely identify physical objects
- Using RFID technology, it *communicates* a binary code

Trident
Spearmint
Menthe verte
WON'T CAUSE CAVITIES • N'ENTRAÎNE PAS DE CARIES
NOW MORE FLAVOUR
LESS INTENSE
MAINTENANT PLUS SAVOUREUSE ET
MOINS INTENSE
12
PIÈCES/MORCEAUX
SUGAR FREE GUM/GOMME SANS SUCRE



EPC Tags



- Passive tags (no battery onboard)
- Memory and Power:
  - Very limited (less than 1024 bits of memory and 4μW)
- Logic Circuitry:
  - Execution of queries, generation of pseudorandom sequences, and integrity checks (CRC).

Gen2 vs. HF-based RFID Standards

Research focused on low-cost RFID technologies:

EPC Gen2

ID 96 - bits

PRNG - 16 bits

Password-protected operations - 32 bits

UHF (865 MHz)

HF

Commercial products

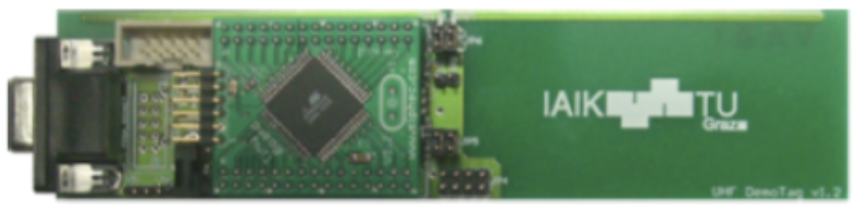
DES,3DES,AES,RSA ...

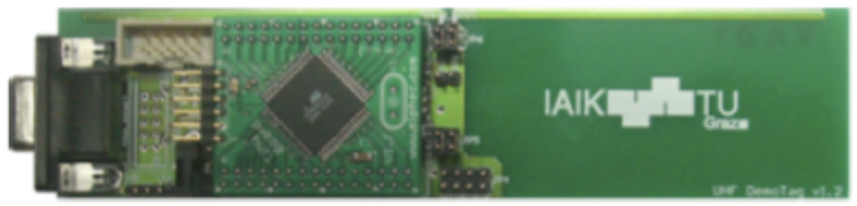
Based on ISO 14443 and ISO 15693

HF (13.56 MHz)

Motivation

- Evaluation of PRNG designs for EPC Gen2 (Industry & Research)
- Weaknesses of the Gen2 protocol if the PRNG output is predictable



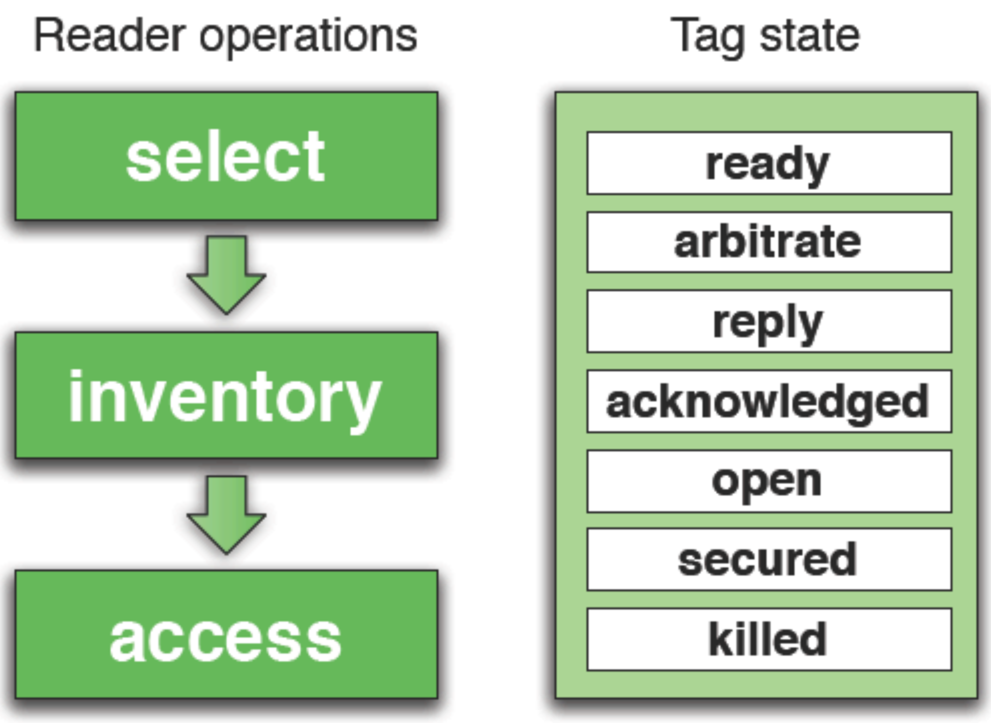


[WPC, 2010] A Practical Implementation Attack on Weak Pseudorandom Number Generator Designs for EPC Gen2 Tags. *Wireless Personal Communications*, Springer, December 2010.

Outline

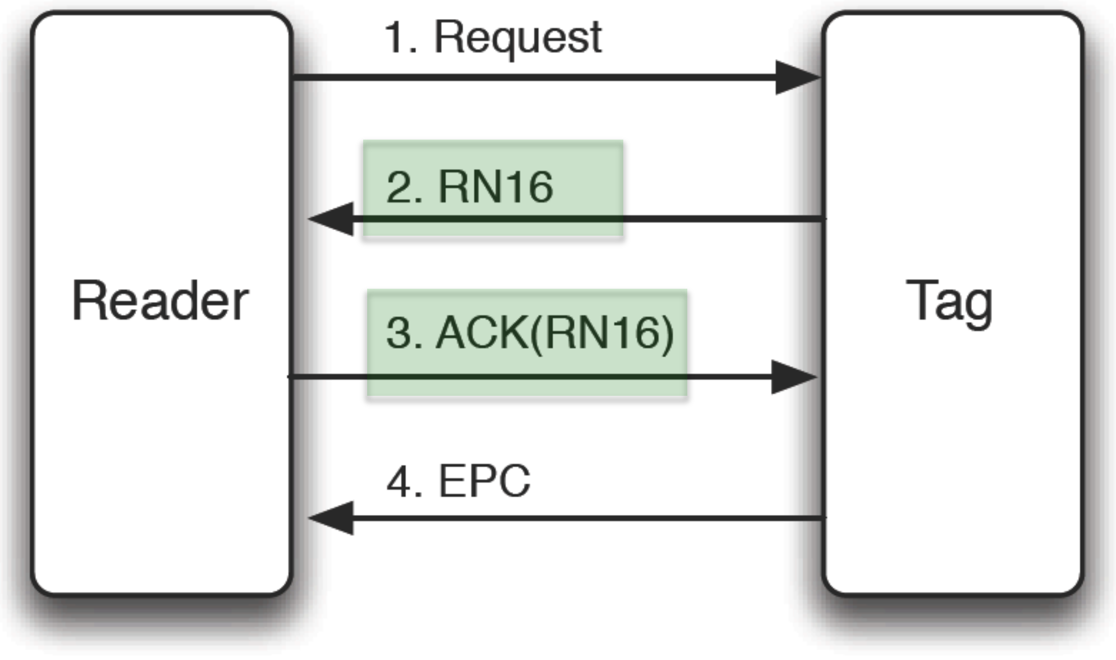
- Introduction
- **EPC Gen2 Protocol**
- **LFSR-based PRNG proposals**
- **Conclusion**

The EPC Gen2 Protocol

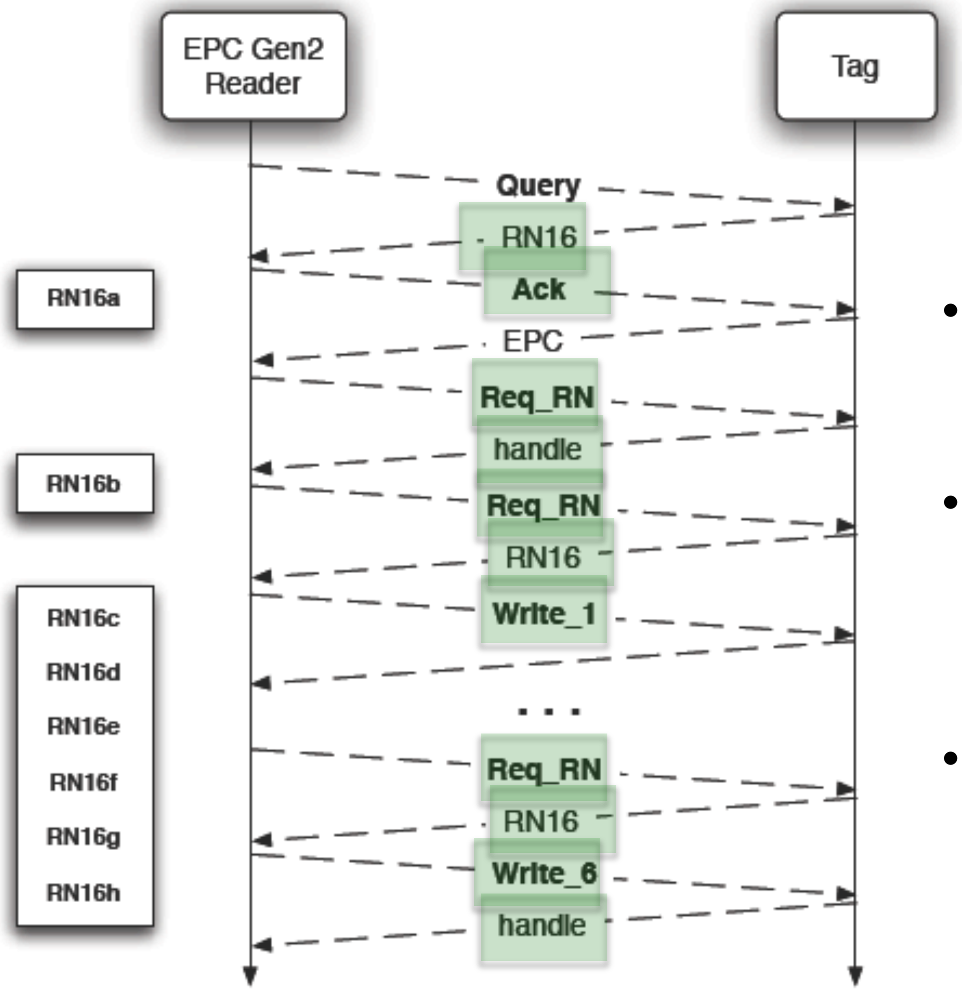


[EPCGlobal, 2010]

Select/Inventory Operation



Access/Open Operation



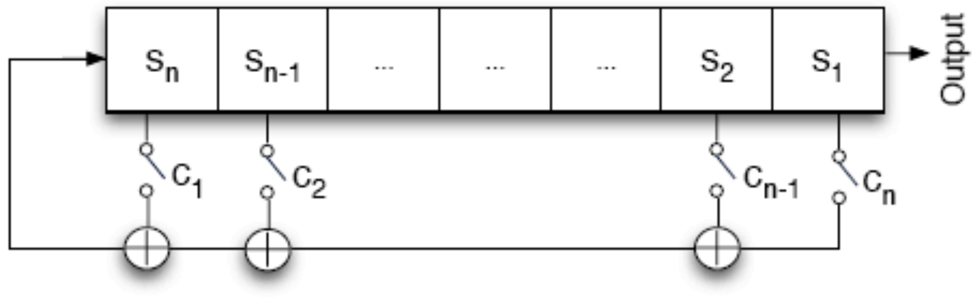
WRITE Command

- After the selection process, the tag is individually identified
- The tag generates several 16 bit nonce series (RN16), used for anti-collision & authentication
- Handle (first RN16) is used to link the command session

Outline

- **Introduction**
- **EPC Gen2 Protocol**
- **LFSR-based PRNG proposals**
- **Conclusion**

LFSR-based PRNGs



- Lightweight hardware implementation
- LFSR already used for Gen2 CRC
- Period of $2^n - 1$ when the feedback polynomial is primitive

$$C(x) = 1 + c_1 s^1 + c_2 s^2 + \cdots + c_n s^n$$

LFSR linearity problem

$$\cdots\ s_{k+1},\quad s_{k+2},\quad s_{k+3},\quad s_{k+4}\quad \cdots\quad s_{k+n},$$

$$s_{k+n+1},\ s_{k+n+2},\ s_{k+n+3},\ s_{k+n+4}\ \cdots\ s_{k+2n},$$

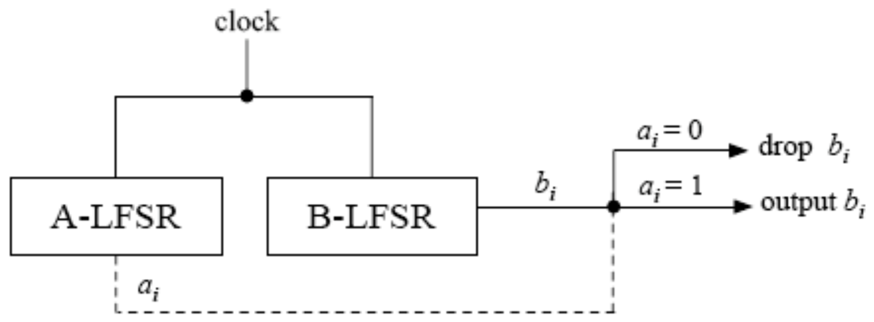
$$s_{k+2n+1},\ s_{k+2n+2},\ s_{k+2n+3}\ \cdots\ s_{k+3n-1}\ \cdots$$

$$\begin{bmatrix} s_{k+1} & s_{k+2} & \cdots & s_{k+n} \\ s_{k+2} & s_{k+3} & \cdots & s_{k+n+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{k+n} & s_{k+n+1} & \cdots & s_{k+2n-1} \end{bmatrix} \begin{bmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \end{bmatrix} = \begin{bmatrix} s_{k+n+1} \\ s_{k+n+2} \\ \vdots \\ s_{k+2n} \end{bmatrix}$$

The feedback polynomial can be determined by simply eavesdropping 2n values

Linearity Avoidance (1/2)

Shrinking generator

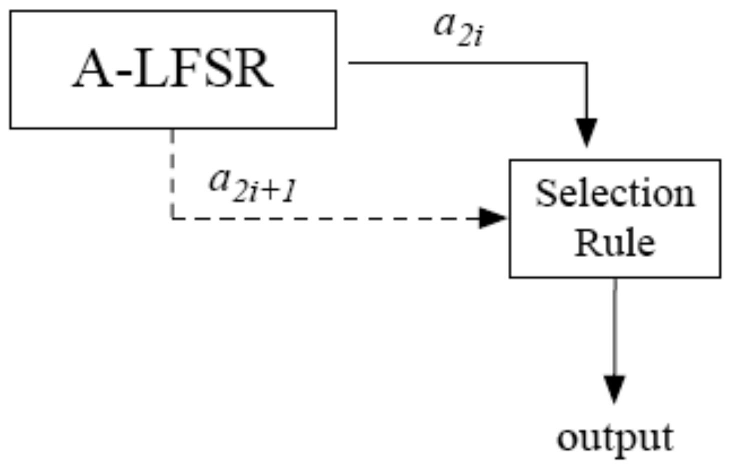


LFSR A generates output bits, LFSR B controlls the output

Shrinking-generator’s output rate varies irregularly

Linearity Avoidance (1/2)

Self-shrinking generator

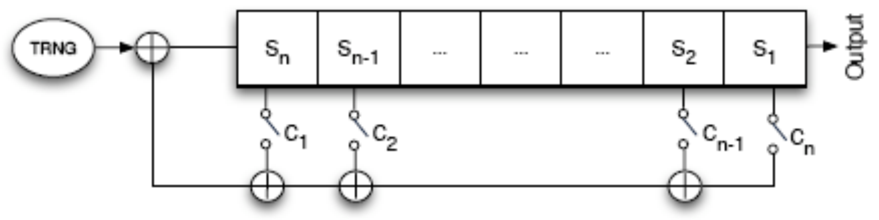


Based on SG but alternating output bits of a single register to generate output bits

Also with irregular otput data rate

Linearity Avoidance (2/2)

Che *et al.* proposal



True Random Number Generation (TRNG)

- *trn* from the thermal noise of two resistors modulating the edge of a sampling clock
- *trn* update each *n* clock times

[Che *et al.*, 2008] Networked RFID Systems and Lightweight Cryptography, Chapter 16, A Random Number Generator for Application in RFID Tags, pp. 279– 287. Springer, 2008.

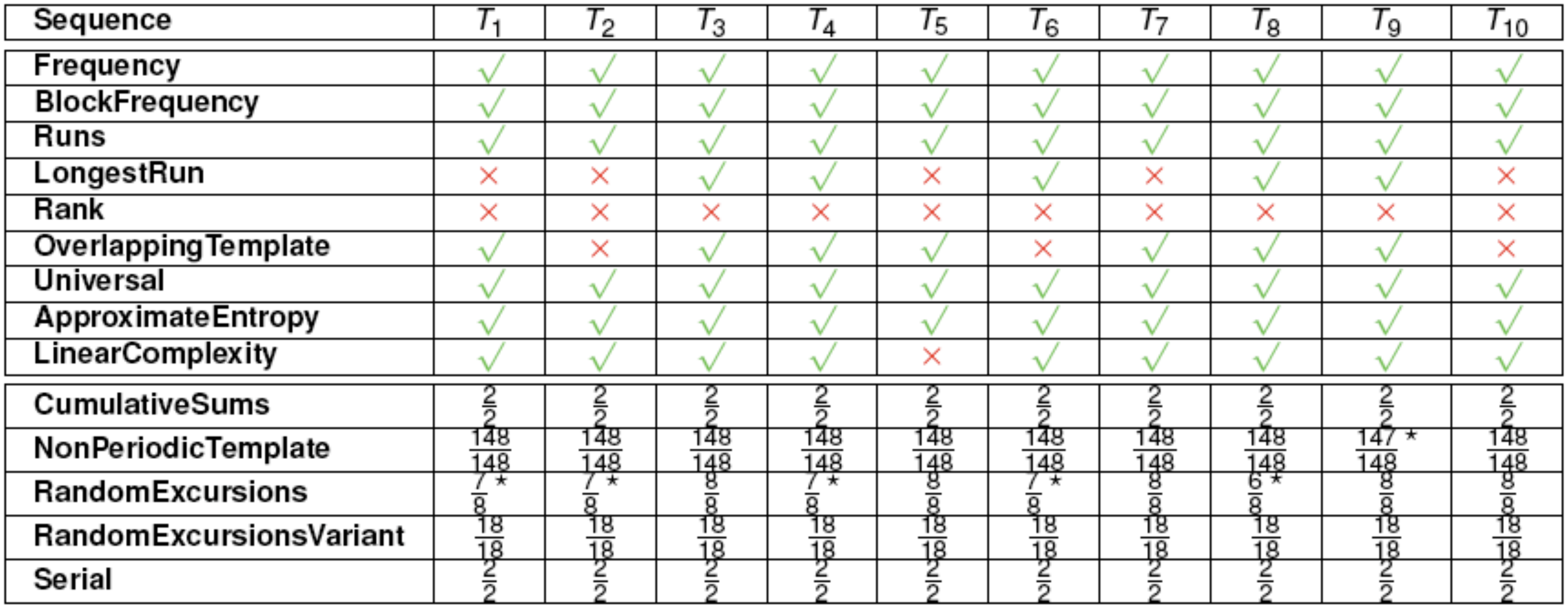
Evaluation of Each Proposal

- Shrinking Generator: ~ 1435 GE, 517 clock cycles at 100KHz
  - EPC requires, at 100KHz, a RN16 in at most 220 cycles
- Che *et al.* Scheme: ~ 500 GE, 50 clock cycles at 100KHz
  - However, predictable with Prob. ~ 1/2 for 160 bits [WLC, 2010]

[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

NIST Statistical analysis

Randomness tests applied to Che et al. PRNG



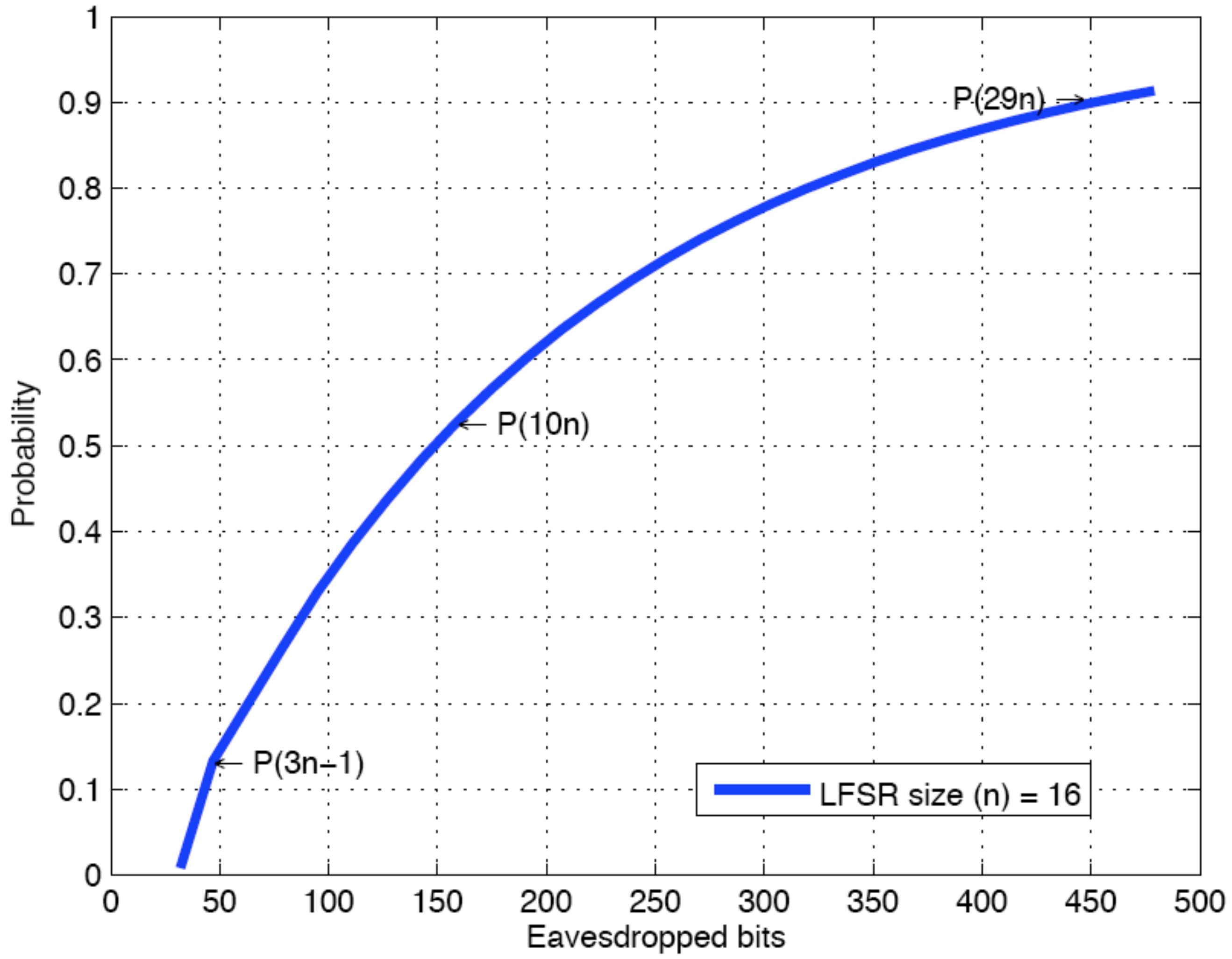
Sequence	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Frequency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BlockFrequency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Runs	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LongestRun	✗	✗	✓	✓	✗	✓	✗	✓	✓	✗
Rank	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
OverlappingTemplate	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗
Universal	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ApproximateEntropy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LinearComplexity	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
CumulativeSums	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$
NonPeriodicTemplate	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{148}{148}$	$\frac{147}{148}\star$	$\frac{148}{148}$
RandomExcursions	$\frac{7}{8}\star$	$\frac{7}{8}\star$	$\frac{8}{8}$	$\frac{7}{8}\star$	$\frac{8}{8}$	$\frac{7}{8}\star$	$\frac{8}{8}$	$\frac{6}{8}\star$	$\frac{8}{8}$	$\frac{8}{8}$
RandomExcursionsVariant	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$	$\frac{18}{18}$
Serial	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$

Statistical test suite for RNGs and PRNGs from the

National Institute of Standards and Technology [NIST 2007]

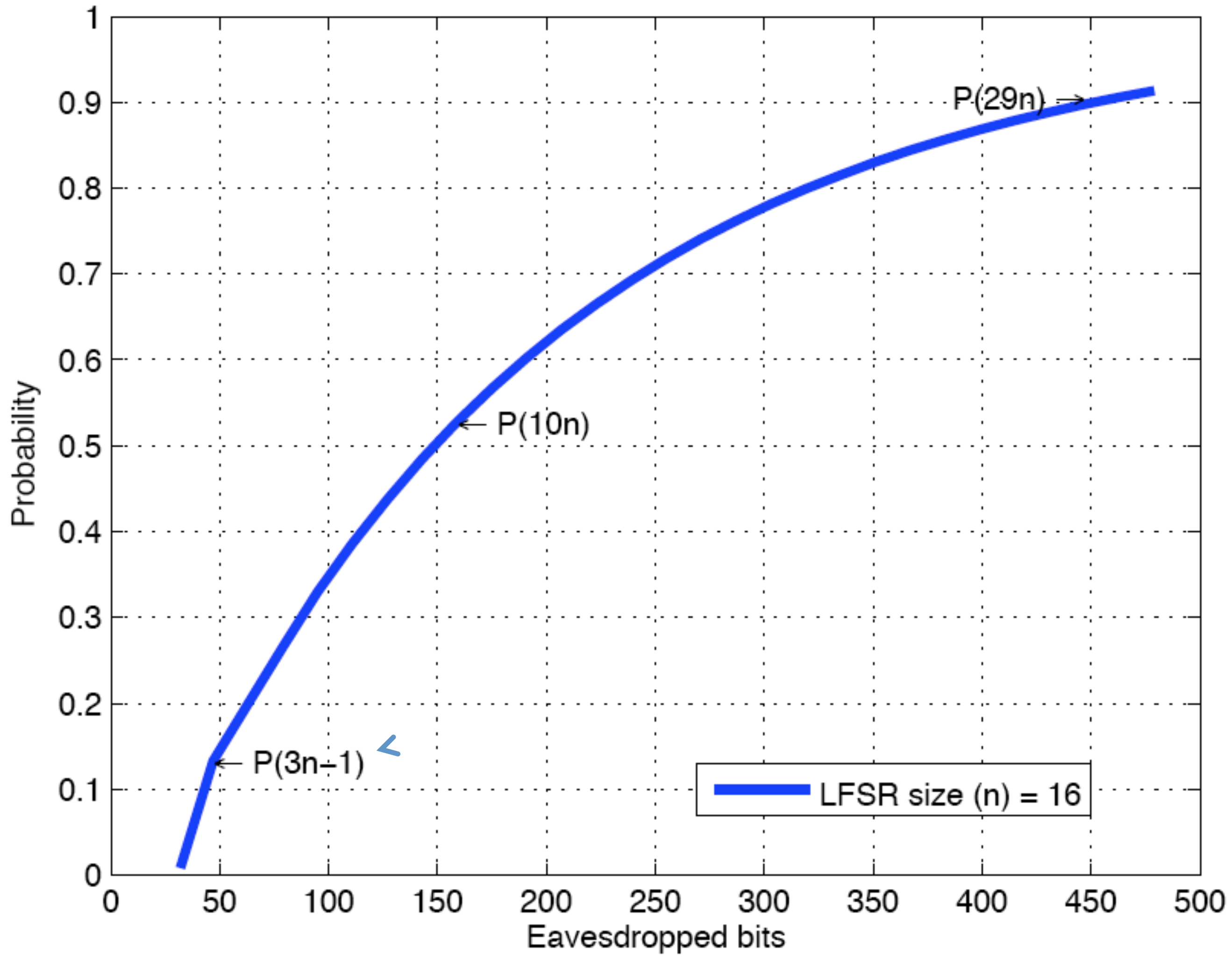
[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Expected Probability of Success



[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Expected Probability of Success



[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Probability of Success

$$P_{(3n-1)}\left(trn_{1,2,3}=0\right)=\boxed{\frac{n+1}{8n}}$$

$$P_{(n=16\rightarrow 47\text{bits})}\left(trn_{1,2,3}=0\right)\approx 13.3\%$$

(4) Prob. that the remainder bits have been affected by exactly three trn

Sequences divided as *2n + (n-1)*

$$\frac{1}{4}\left(\frac{1}{n}\right)+\frac{1}{8}\left(\frac{n-1}{n}\right)$$

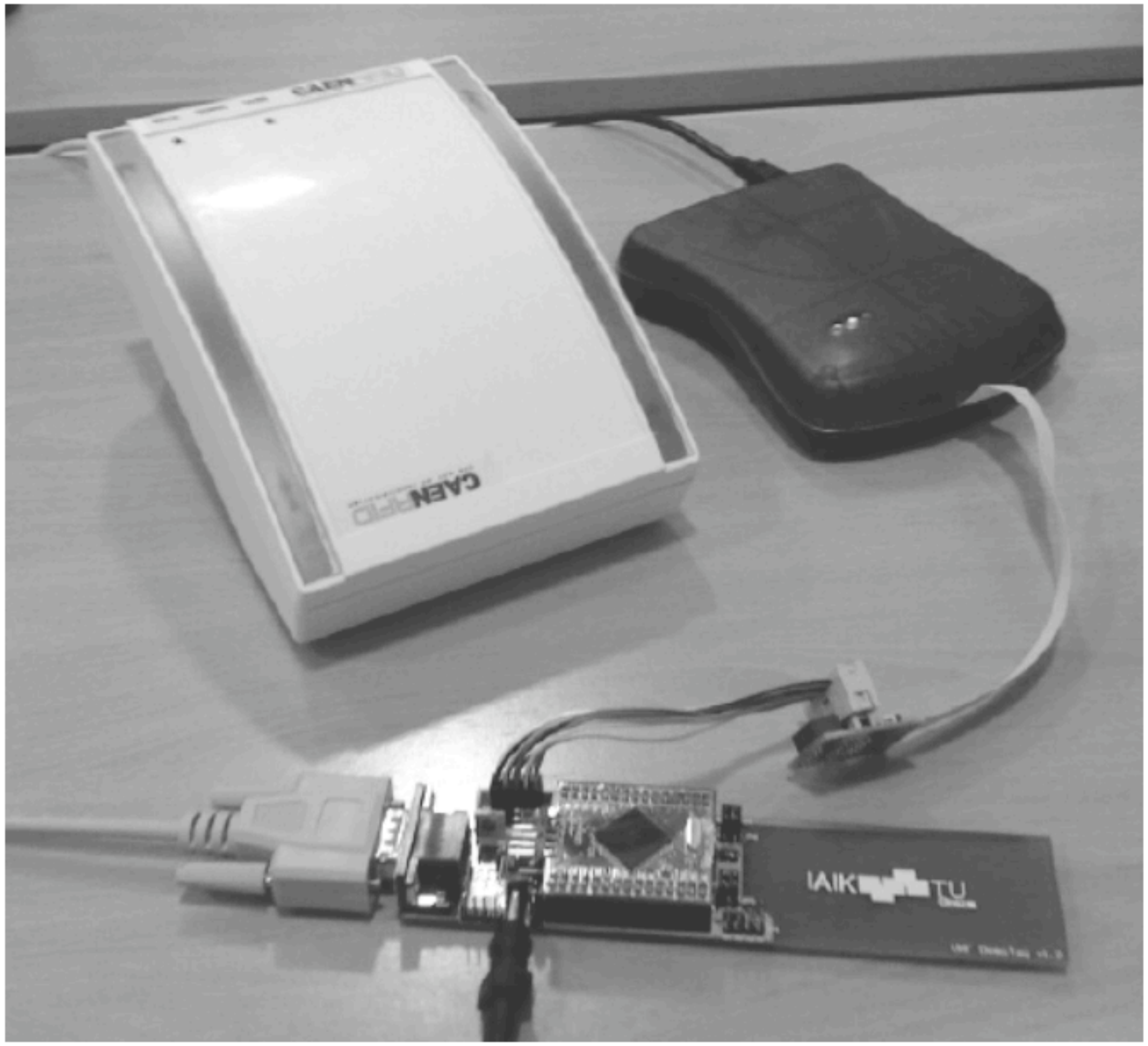
(2) Probability that the two *trn* used in that sequence are exactly zeros

(1) Given a sequence *s*, s.t. $|s| = 2n$, prob. that *s* has been affected by exactly two trn

(3) Probability that the three *trn* used in that sequence are exactly zeros

[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Testing the Attack (1/2)



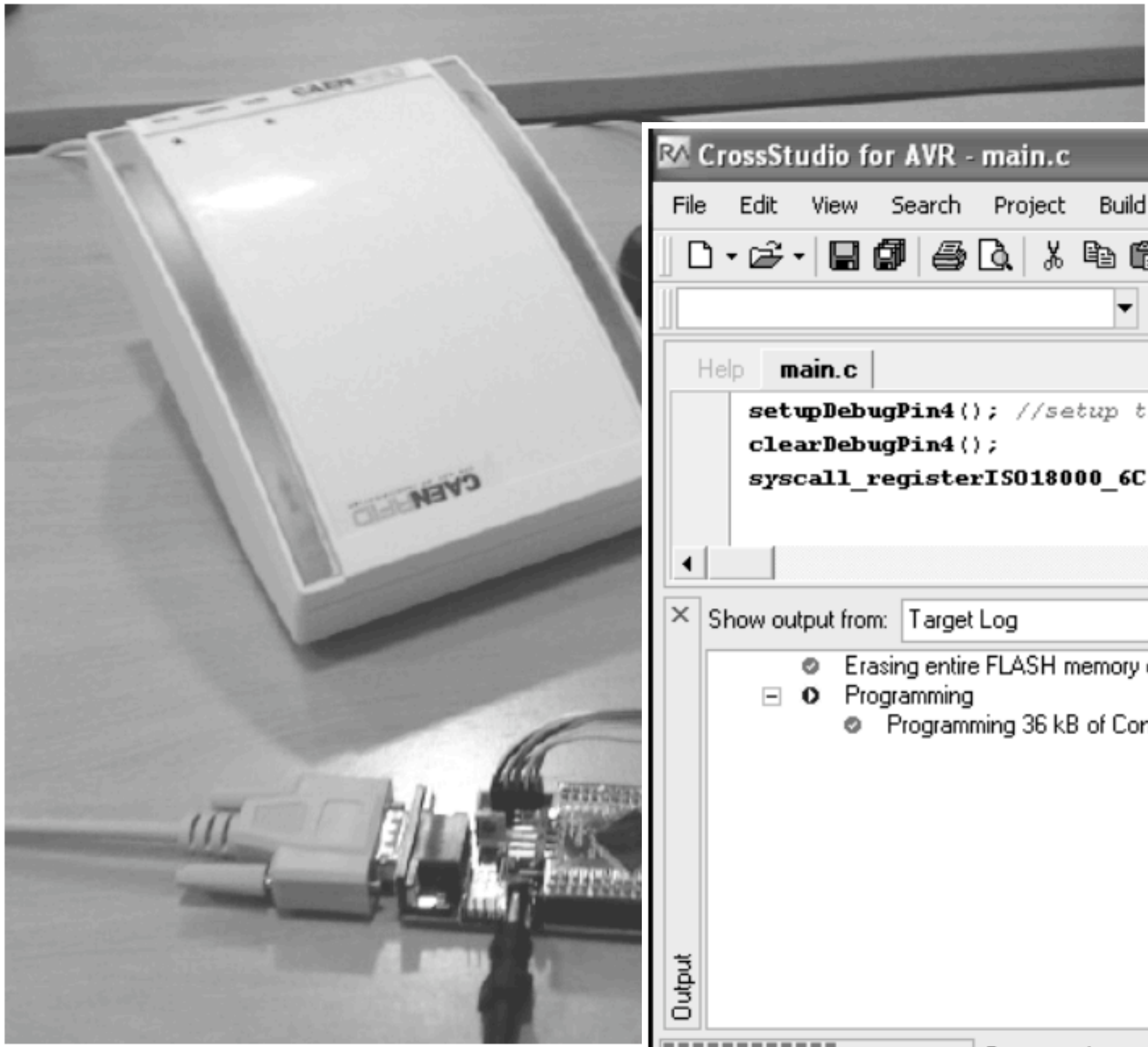
```
rigger pin4 for debug

(ISO_18000_6C_protocol);
```

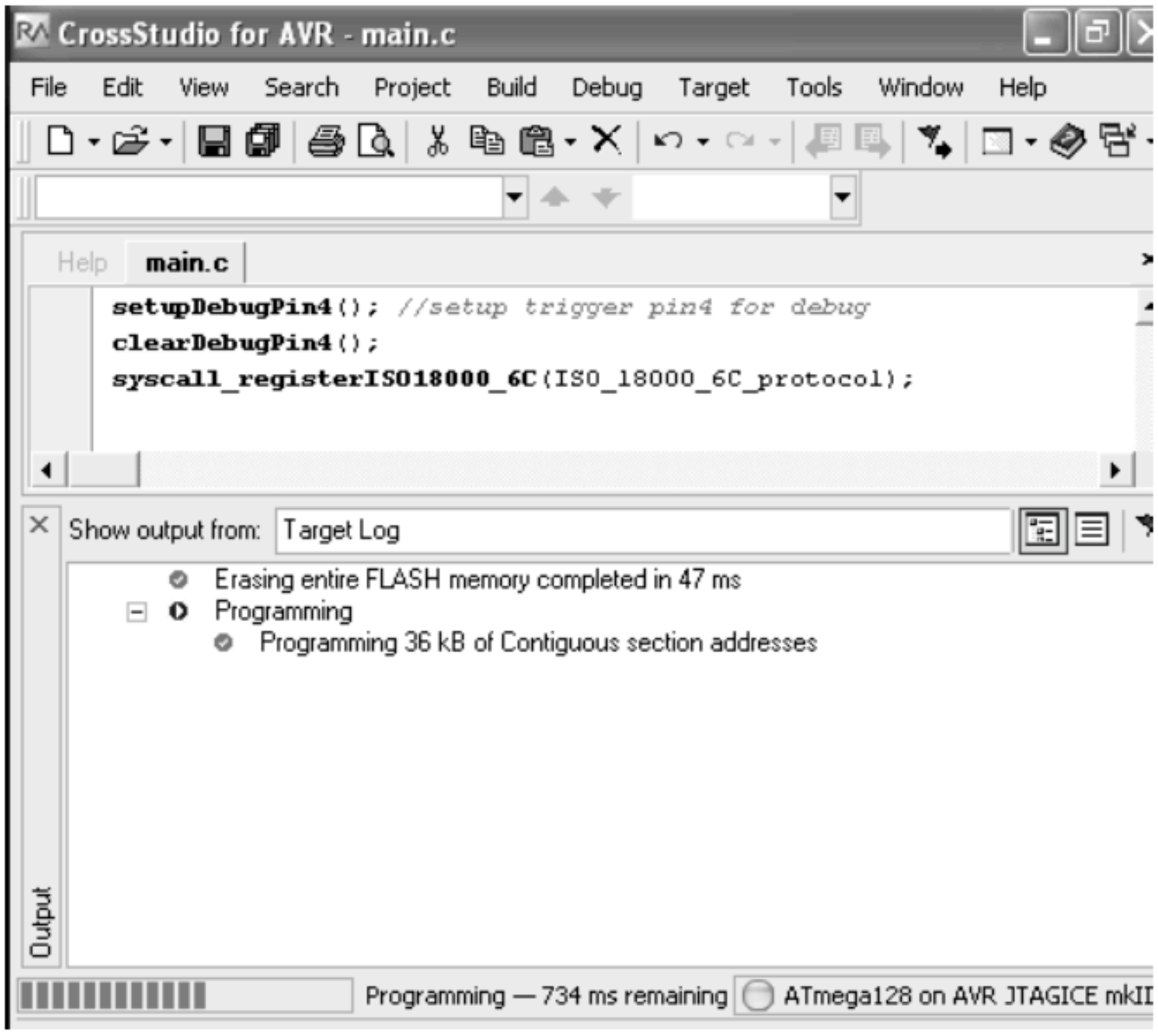
ompleted in 47 ms

tiguous section addresses

734 ms remaining | ATmega128 on AVR JTAGICE mkII

Testing the Attack (1/2)




CrossStudio for AVR - main.c
File Edit View Search Project Build Debug Target Tools Window Help
Help main.c
Output

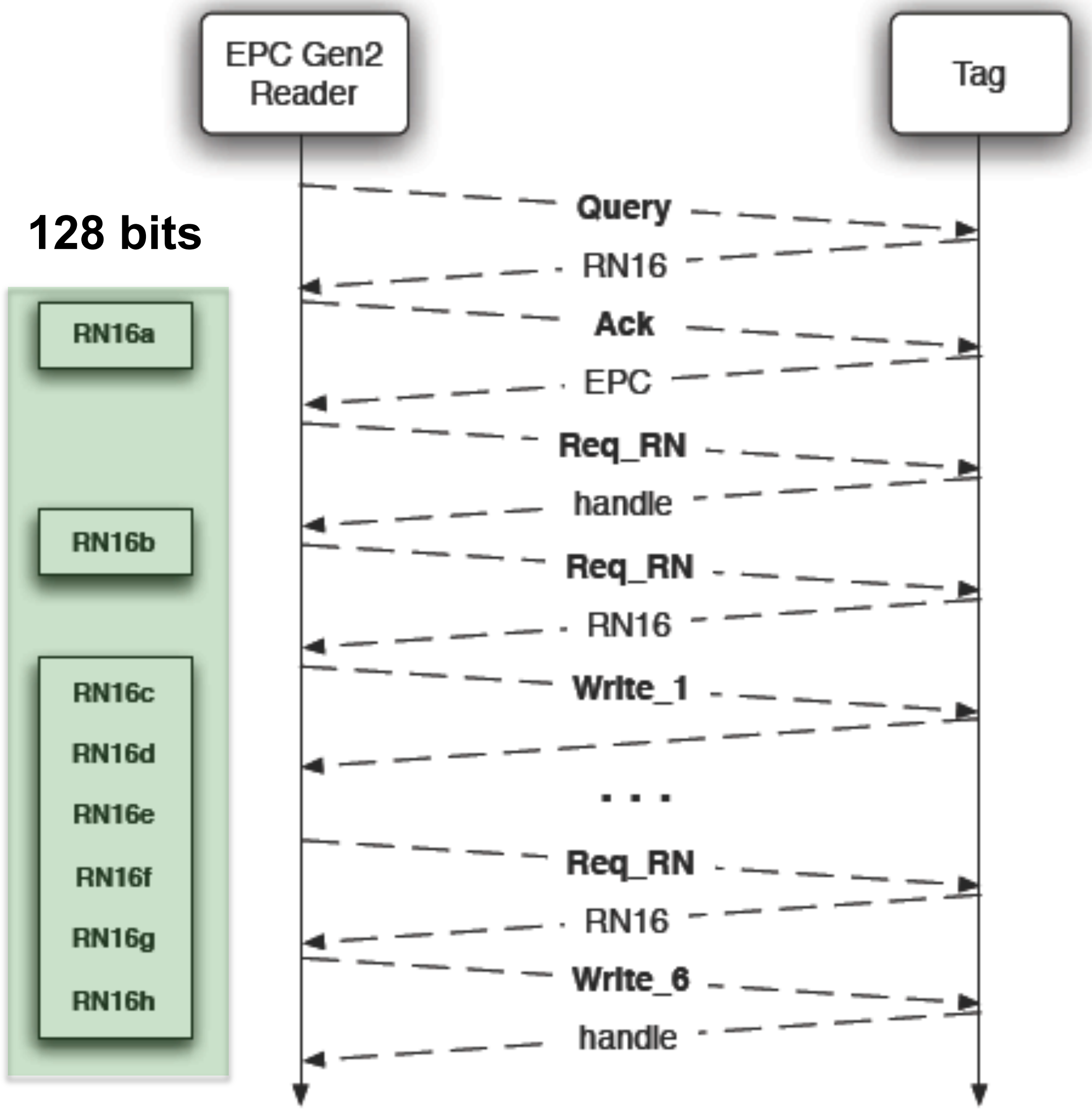
```
setupDebugPin4(); //setup trigger pin4 for debug
clearDebugPin4();
syscall_registerISO18000_6C(ISO_18000_6C_protocol);
```

Show output from: Target Log

```
Erasing entire FLASH memory completed in 47 ms
Programming
    Programming 36 kB of Contiguous section addresses
```

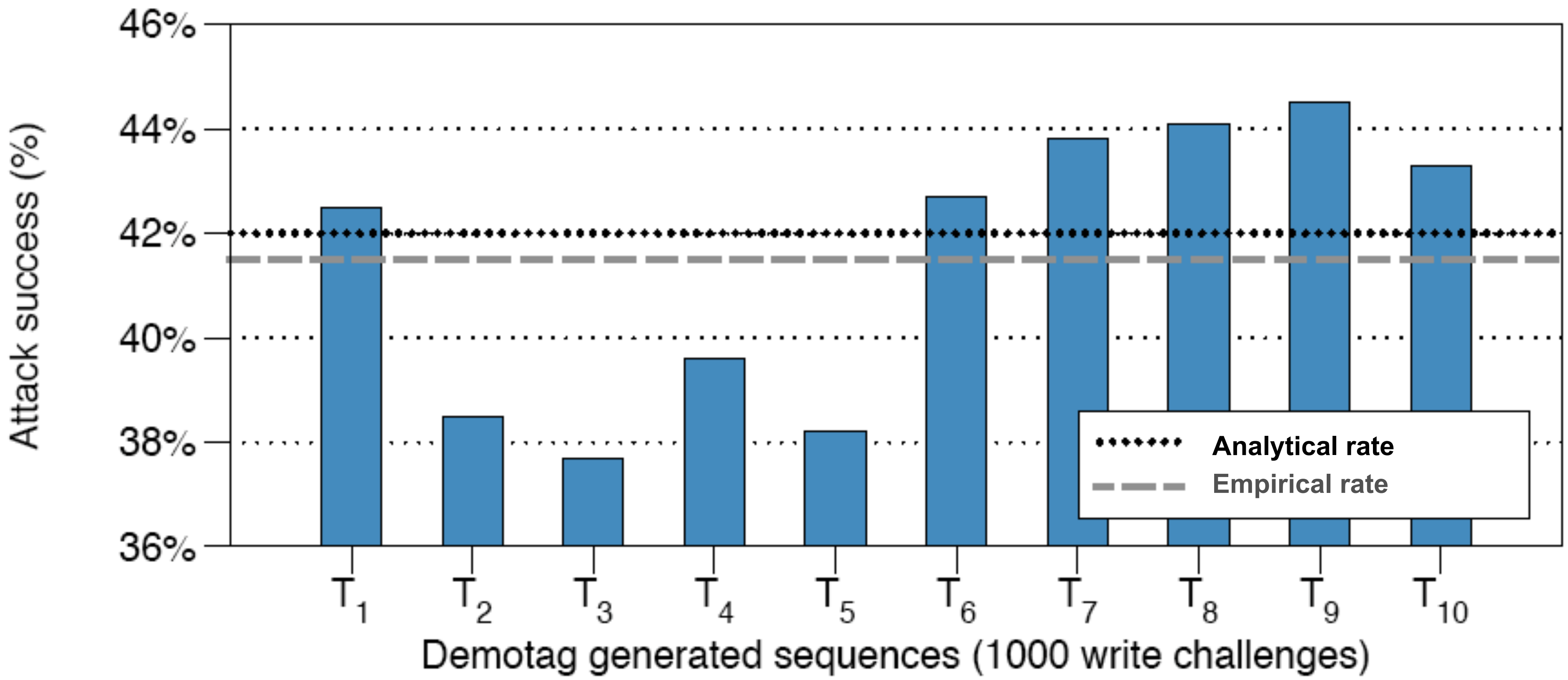
Programming — 734 ms remaining | ATmega128 on AVR JTAGICE mkII

Testing the Attack (2/2)



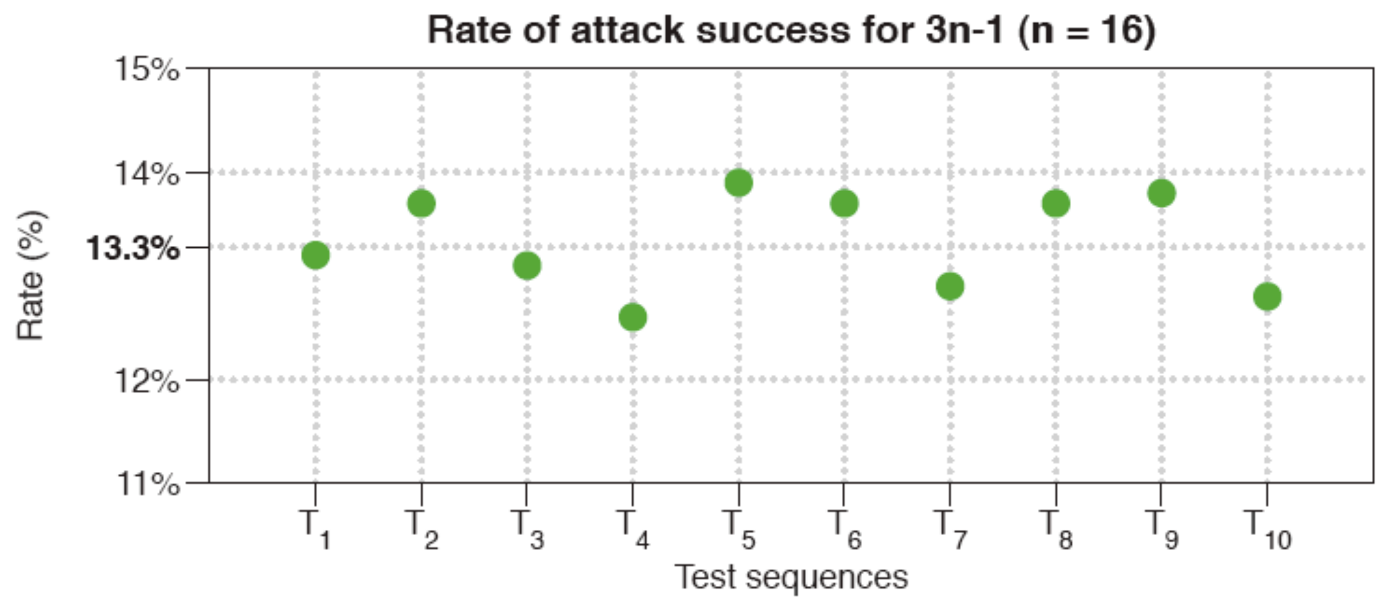
```
=> QUERY ...
=> ACK ...
=> Req_RN RN:14438
=> Req_RN RN:44282
=> Write (data) 27698
=> Req_RN RN:44282
=> Write (data) 47380
=> Req_RN RN:44282
=> Write (data) 44282
=> Req_RN RN:44282
=> Write (data) 60868
=> Req_RN RN:44282
=> Write (data) 32656
=> Req_RN RN:44282
=> Write (data) 34674
```

Success Rate



[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Success Rate



Implementation for 16 cells LFSR (n=16)

Ten test sequences ($T_{1..10}$) of approx. 340 MB

Different *trn* source, seed and primitive $C(x)$

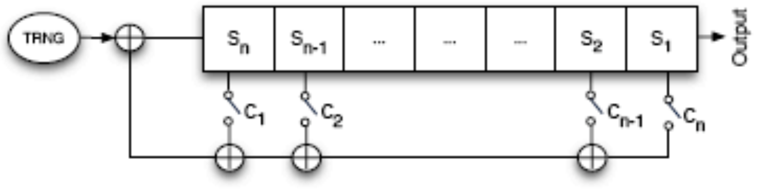
[WLC, 2010] Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags, Financial Cryptography and Data Security 2010 Workshops, LNCS, Springer, January, 2010.

Outline

- Introduction
- EPC Gen2 Protocol
- LFSR-based PRNG proposals
- **Work-in-Progress**
- **Conclusion**

Work-in-Progress

Attack on PRNG proposal:



Exploiting linearity of LFSR

$P_{(n=16\rightarrow 47\text{bits})}\left(trn_{1,2,3}=0\right) \approx 13.3\%$

Detect vulnerabilities

- LFSR, LCG, ...
- NIST, Diehard, Ent, ...
- Weak output

Analyze sequences

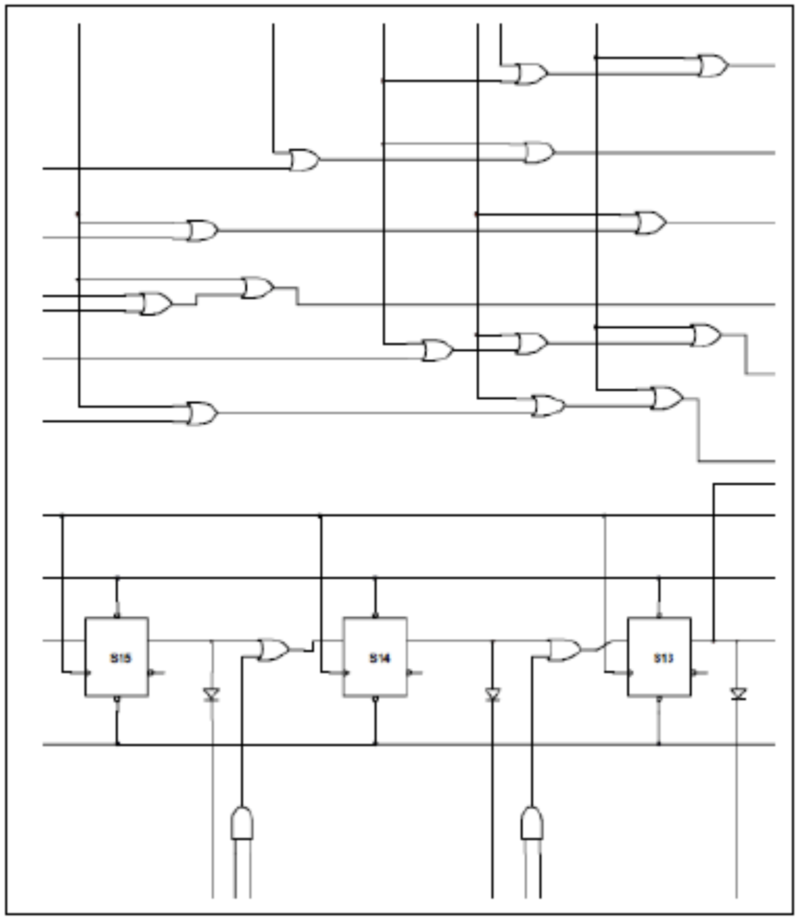
- Demotag implementation
- Software simulation

Evaluate weaknesses

- Eventual synchronization
- Reveal passwords
- Tamper data

Work-in-Progress Status

Security improvements implemented in hardware:



Hardware complexity

- Logic Gate equivalence
- Evaluates system complexity
- Area of implementation

Execution time

- Depends on internal clock
- *Unlimited* energy

Power consumption eval.

- Number of GE
- No clock cycle should consume excessive power
- $> 4\mu W$ (EPC Gen2)

Outline

- **Introduction**
- **EPC Gen2 Protocol**
- **LFSR-based PRNG proposals**
- **Conclusion**

Conclusion

EPC Gen2: *Pseudorandom Number Generator* as main security tool

Che et al. PRNG is vulnerable due to Linear Feedback Shift Register linearity

- Probability of $\approx \frac{1}{8}$ (for 3n-1 bits) to predict the output

Proposed PRNG based on Multiple Polynomials approach

- Also based on combination of LFSR and *trn*
- Linearity avoidance by switching primitive polynomials
- Compatible with Gen2 specifications
- Suitable statistical behavior

Work-in-Progress Status

- Power consumption evaluation
    - Depends on CMOS technology used
    - Number of GE
    - No clock cycle should consume excessive power
    - $> 4\mu W$ (EPC Gen2)

- Security evaluation
    - Find the probability to predict the output
    - Regarding adversary capabilities

A MP-LFSR based Pseudorandom Number Generator for EPC Gen2 Systems

Mitacs Workshop on Network Security & Cryptography