Serving society
Stimulating innovation
Supporting legislation

Improved Cryptanalysis of the DECT Standard Cipher

Iwen Coisel,

Ignacio Sanchez

CHES 2015 – Saint-Malo, 15/09/2015

European Commission

Our Results in One Slide

- Known-Plaintext Attack against the DECT Standard Cipher (DSC)
- Inspired by the Nohl-Tews-Weinman (NTW) attack[1] but more efficient

  → The attack needs 4 time less plaintext

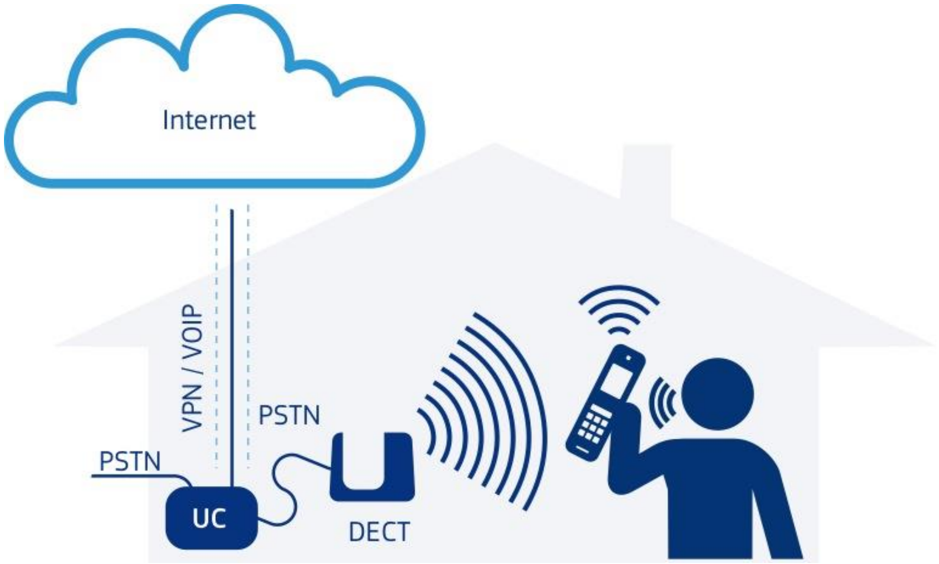
- Attack performed against actual communications
- Attack still feasible in non-ideal conditions (plaintext recovery 90%)

---

[1] K. Nohl, E. Tews, R.P. Weinmann, Cryptanalysis of the DECT Standard Cipher. In Fast Software Encryption. Pp. 1-18. Springer 2010

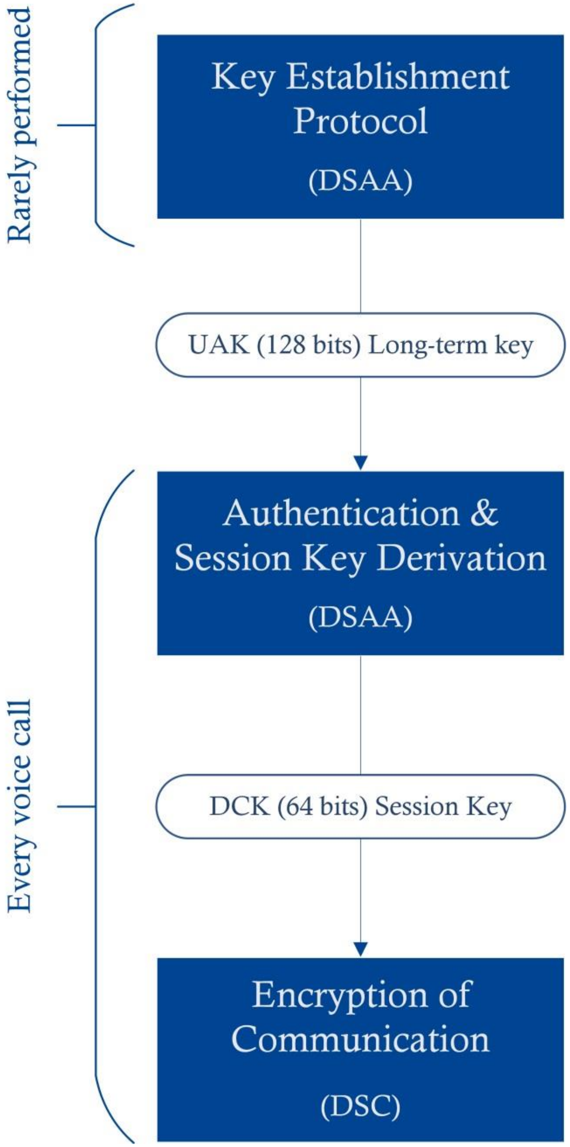
Generalities about the DECT Standard

Traditional Usage vs Modern Usage



- **Residential cordless phones connected to PSTN**
- **Enterprise cordless phones connected to PBX or Unified Communication Systems**
- **As residential cordless phones connected to UC.**
  - VoIP + PSTN hybrids
  - New generation of home UC, integrating WiFi + DECT

Overview of the Cryptographic Mechanisms



Rarely performed

Key Establishment Protocol

(DSAA)

UAK (128 bits) Long-term key

Authentication & Session Key Derivation

(DSAA)

DCK (64 bits) Session Key

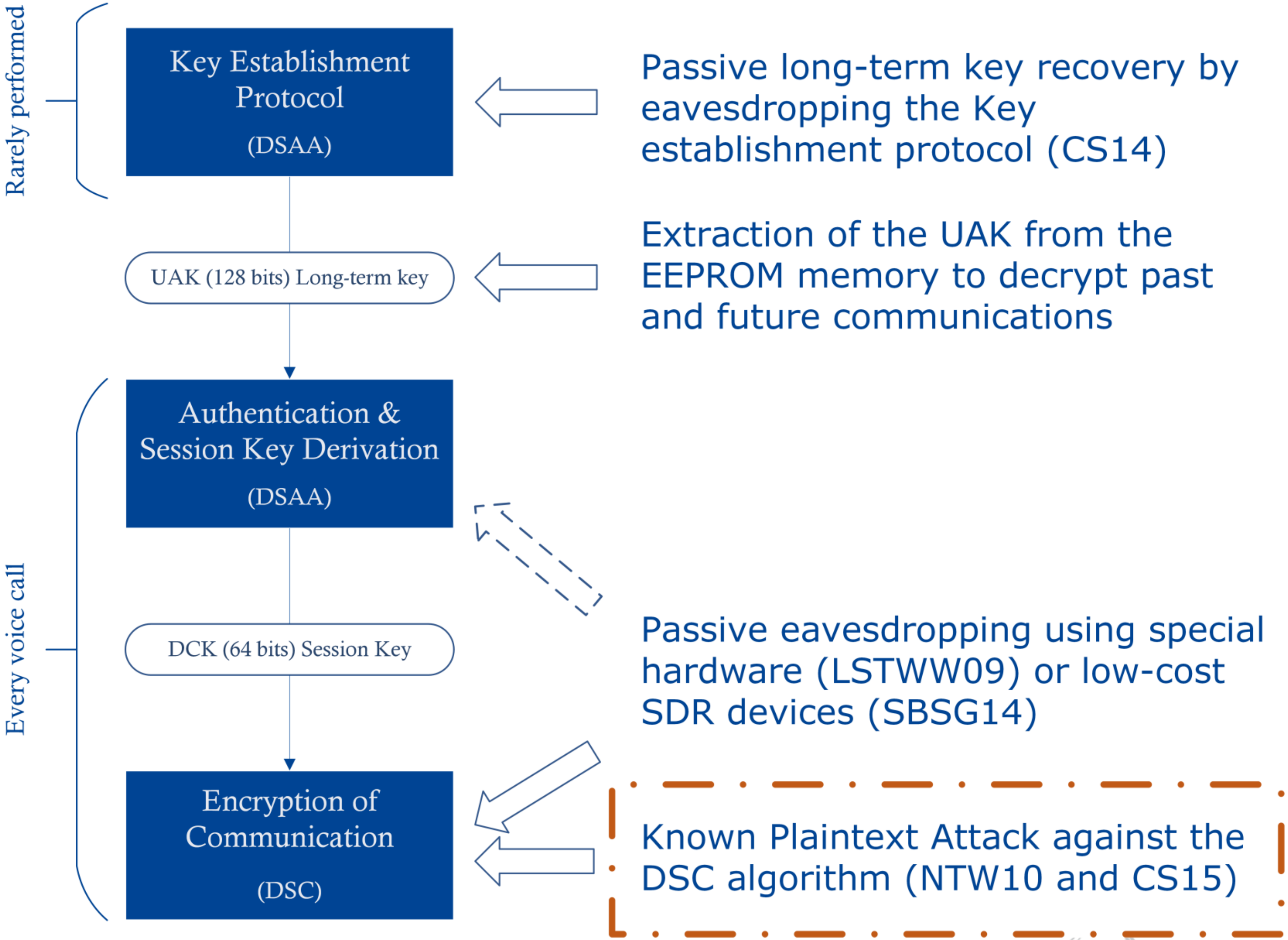
Encryption of Communication

(DSC)

Every voice call

- **DECT Standard Authentication Algorithm (DSAA)**
  - Block cipher
  - 192 bits input / 128 bits output
- **User Authentication Key (UAK)**
  - 128 bits
  - Obtained with A_{21} (DSAA based)
- **DSC Cipher Key (DCK)**
  - 64 bits
  - Obtained with A_{12} (DSAA based)
- **DECT Standard Cipher (DSC)**
  - Asynchronous cipher with 4 Gallois LFSRs
  - Input: 64 bit DCK + 35 bits IV
  - Output: 720 bits of keystream

Overview of the Known Attacks



Rarely performed

Key Establishment Protocol

(DSAA)

UAK (128 bits) Long-term key

Every voice call

Authentication & Session Key Derivation

(DSAA)

DCK (64 bits) Session Key

Encryption of Communication

(DSC)

Passive long-term key recovery by eavesdropping the Key establishment protocol (CS14)

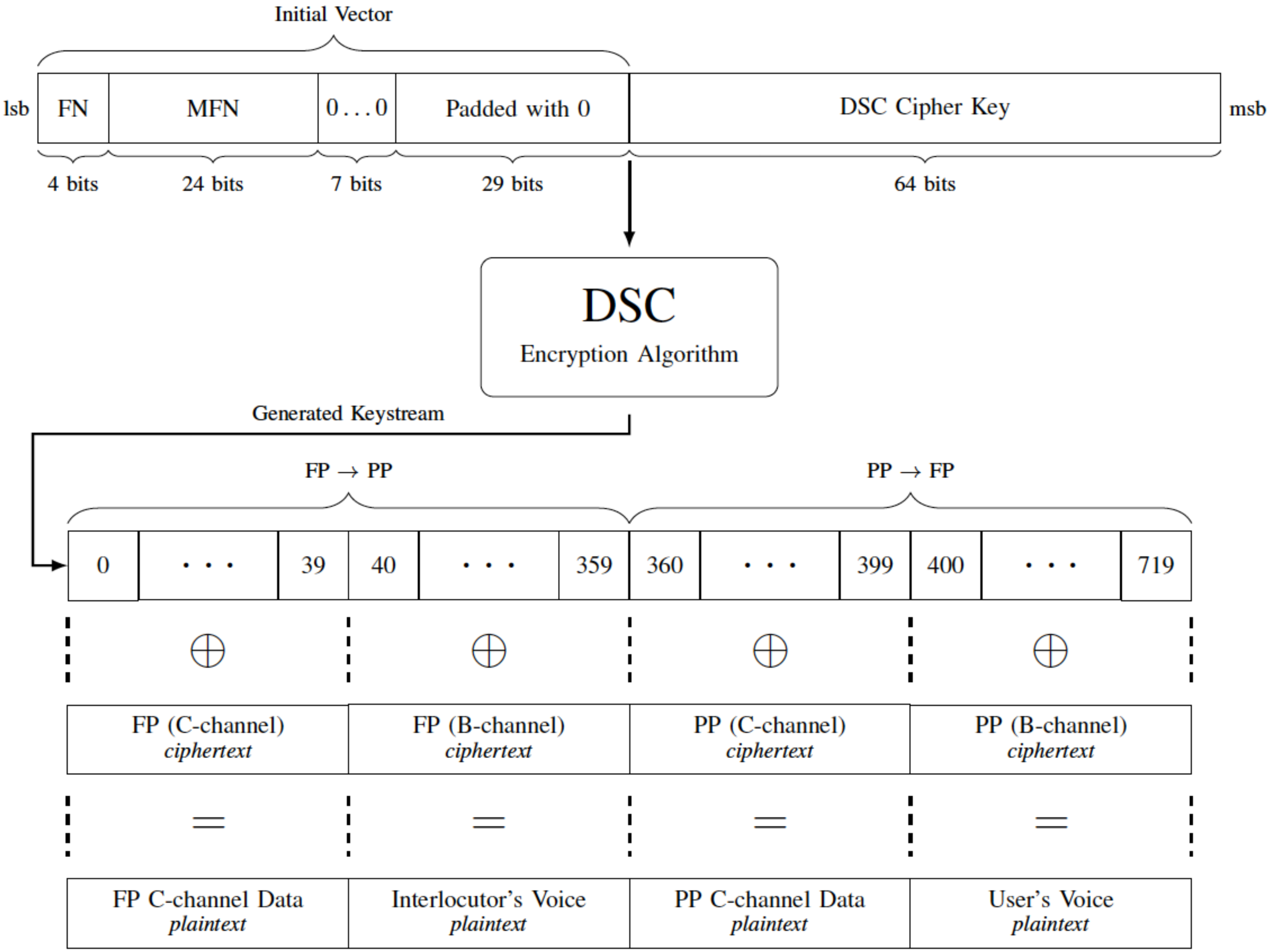
Extraction of the UAK from the EEPROM memory to decrypt past and future communications

Passive eavesdropping using special hardware (LSTWW09) or low-cost SDR devices (SBSG14)

Known Plaintext Attack against the DSC algorithm (NTW10 and CS15)

Focus on the DECT Stream Cipher

Encryption / Decryption Procedure in Details



Initial Vector

lsb | FN | MFN | 0…0 | Padded with 0 | DSC Cipher Key | msb

4 bits | 24 bits | 7 bits | 29 bits | 64 bits

DSC
Encryption Algorithm

Generated Keystream

FP → PP | PP → FP

0 | · · · | 39 | 40 | · · · | 359 | 360 | · · · | 399 | 400 | · · · | 719

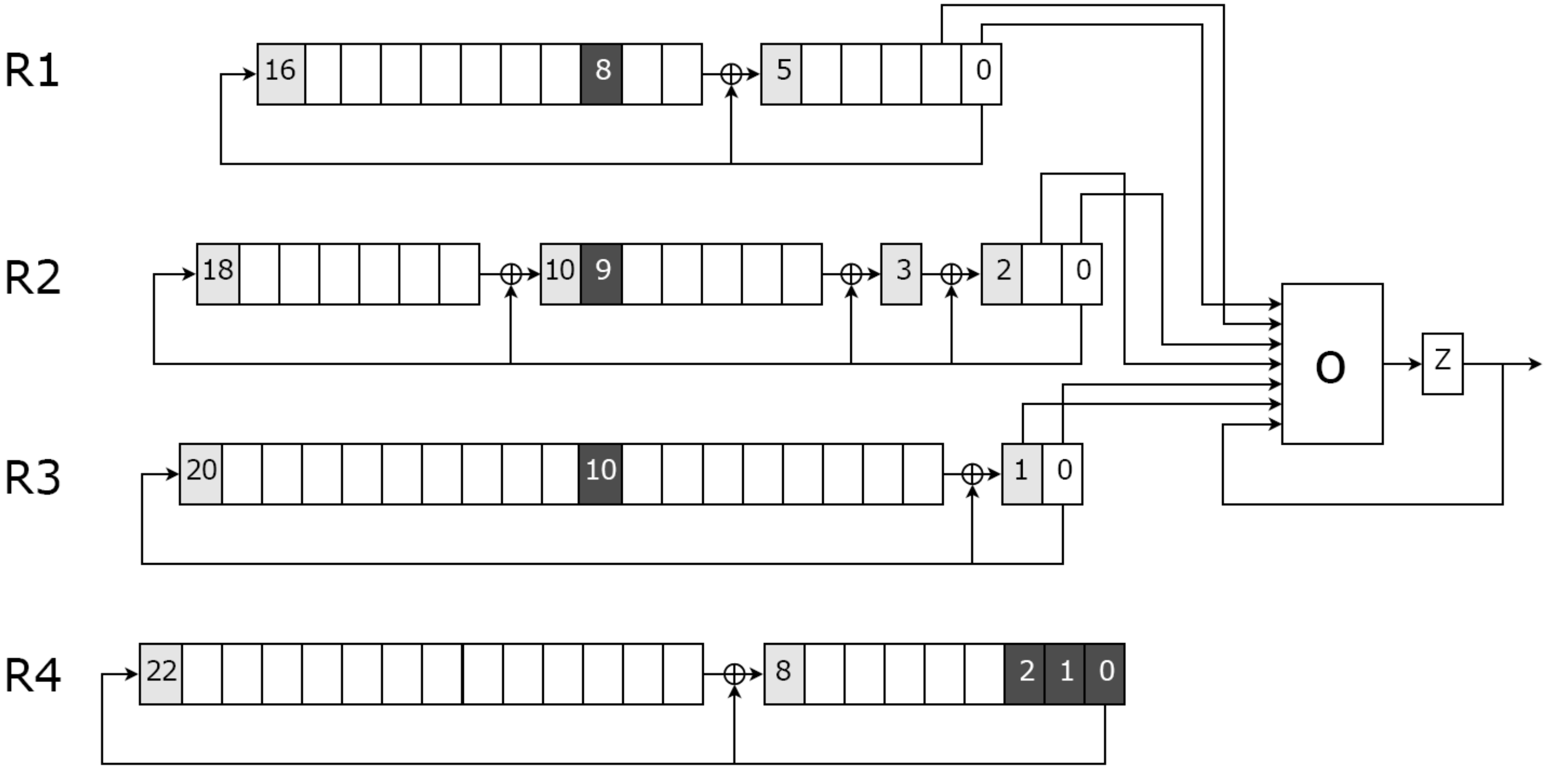
⊕ ⊕ ⊕ ⊕

FP (C-channel) *ciphertext* | FP (B-channel) *ciphertext* | PP (C-channel) *ciphertext* | PP (B-channel) *ciphertext*

= = = =

FP C-channel Data *plaintext* | Interlocutor's Voice *plaintext* | PP C-channel Data *plaintext* | User's Voice *plaintext*

The DECT Stream Cipher



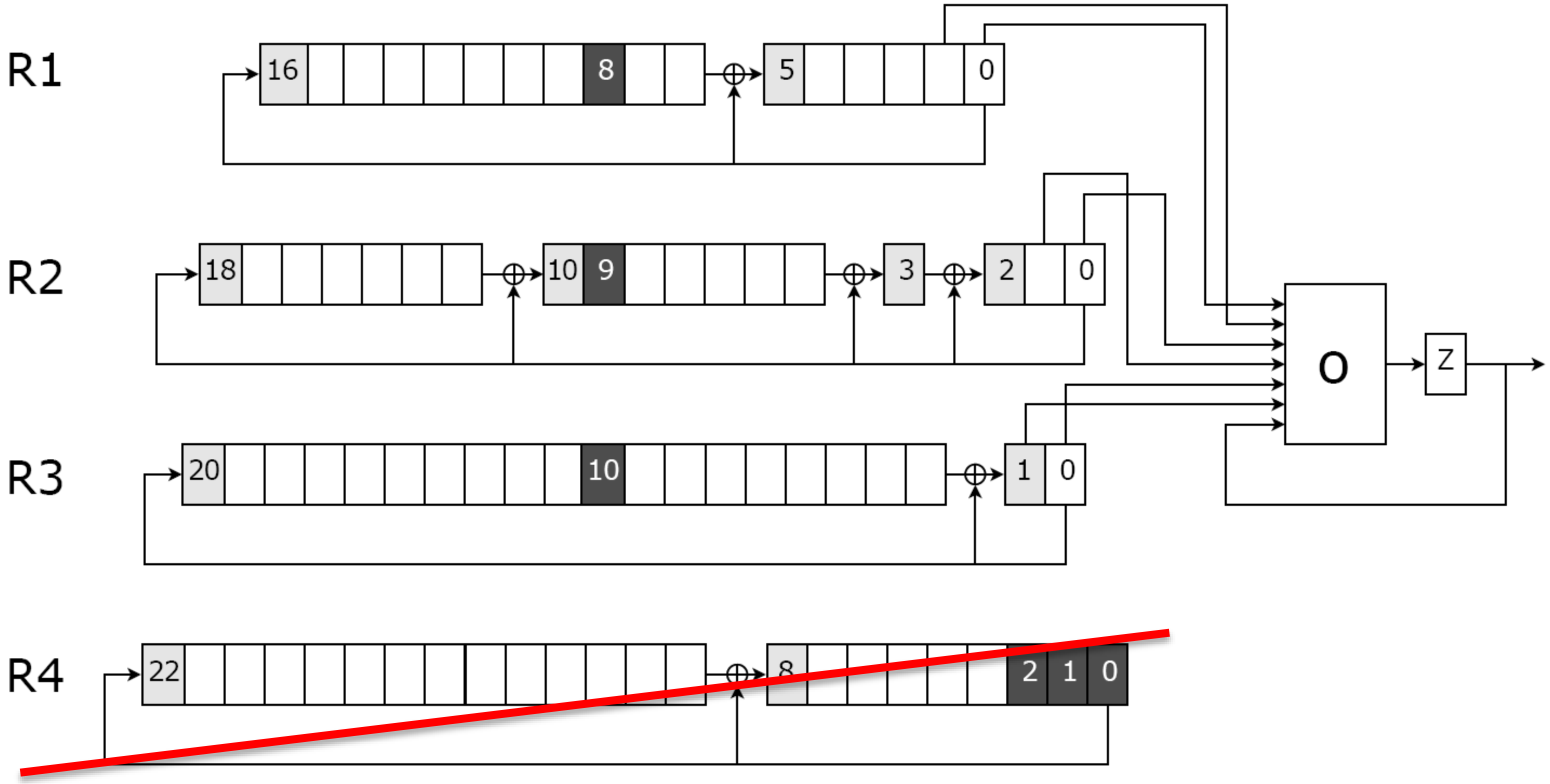
Irregular clocking of the registers:

- $R1 = 2 + (x_{4,0} \oplus x_{2,9} \oplus x_{3,10})$
- $R2 = 2 + (x_{4,1} \oplus x_{1,8} \oplus x_{3,10})$
- $R3 = 2 + (x_{4,2} \oplus x_{1,8} \oplus x_{2,9})$
- $R4 = 3$

Output Combiner:

$O(S,z) = x_{1,1}x_{1,0}z \oplus x_{2,0}x_{1,1}x_{1,0} \oplus x_{1,1}z \oplus x_{2,1}x_{1,0}z$
$\oplus x_{2,1} \oplus x_{2,1}x_{2,0}x_{1,0} \oplus x_{3,0}z \oplus x_{3,0}x_{1,0}z \oplus x_{3,1} \oplus x_{3,1}z$
$\oplus x_{3,0}x_{2,0}x_{1,0} \oplus x_{1,1}x_{1,0} \oplus x_{2,0}x_{1,1} \oplus x_{3,1}x_{1,0}$

The DECT Stream Cipher



Irregular clocking of the registers:

- R1 =
- R2 =
- R3 =
- ~~R4 = 3~~

Randomly and independently clocked 2 or 3 times

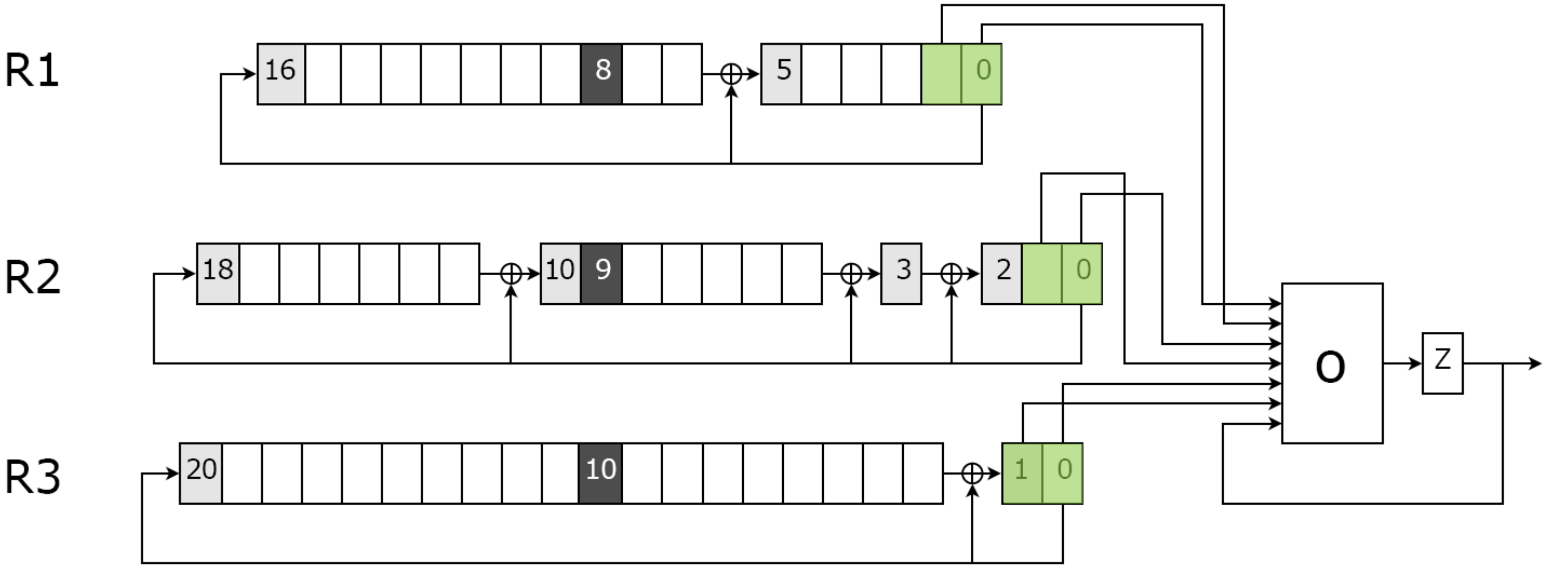
Output Combiner:

$O(S,z) = x_{1,1}x_{1,0}z \oplus x_{2,0}x_{1,1}x_{1,0} \oplus x_{1,1}z \oplus x_{2,1}x_{1,0}z$

$\oplus\, x_{2,1} \oplus x_{2,1}x_{2,0}x_{1,0} \oplus x_{3,0}z \oplus x_{3,0}x_{1,0}z \oplus x_{3,1} \oplus x_{3,1}z$

$\oplus\, x_{3,0}x_{2,0}x_{1,0} \oplus x_{1,1}x_{1,0} \oplus x_{2,0}x_{1,1} \oplus x_{3,1}x_{1,0}$

Setup and Notations



Initialisation of the DSC

- Loading of the IV and then the key in the registers clocking one time after each bit
- 40 “empty” rounds with irregular clocking where the keystream bits are discarded

Status of the DSC, 6 bits (in green) given as input to the output combiner. It is defined by:

- A number of rounds or a triplet of clocks
- A key and / or an IV

S_c(Key,IV) S_c(0,IV) S_c(Key,0)
S_l(Key,IV) S_l(0,IV) S_l(Key,0)

Description of our Known Plaintext Attack

Basic Idea of the Attack

We have re-used the core idea of the NTW attack:

- Each bit of each register for a given number of clocks can be defined as a linear equation of the bits of the key and the bits of the initial vector
- **Goal:** guess the status of the DSC for a known triplet of clocks

  → 6 linear combinations of the bits of the key
- Recover the status for a sufficient amount of clocks in order to determine enough linear equations (≈ 20 – 30 equations)
- Brute-force the remaining bits ($64 - nb_{equations}$)

Guessing Correctly a Status 1/2

What do we know?

- Several thousands of couple (IV, Keystream $(z_0,\dots,z_{719})$)
- S_c(0,IV) that can be computed for any triplet of clocks c
- $O(S\_l(Key,IV), z_{l-1}) = z_l$ for $l \in \{0,719\}$ [*Eqn(st,IV,l)*]

What do we want?

- S_c(Key,0) for several triplets of clocks

If the triplet of clock c is correct for a given round l then:

1. $S\_l(Key,IV) = S\_c(Key,IV) = S\_c(Key,0) \oplus S\_c(0,IV)$
2. $S\_c(Key,0) \in CST = \{st \mid st^* = st \oplus S\_c(0,IV)$ verify *Eqn(st*,IV,l)*$\}$

All the other status have 50% of chances to be in this subset

Guessing Correctly a Status 2/2

Last useful fact:

The number of clocks for a given round is distributed according to a shifted polynomial distribution of mode 2,5l + 100

Example: for round 1 the most probable number of clock is 102,5

How do we use these facts?

Let c = (102,102,102) be the expected triplet of clock for the first round

For each IV we determine:

- S_c(0,IV)
- CST = {st | st ⊕ S_c(0,IV) verify *Eqn(st,IV,l)*}

It can be seen as a Bernouilli trial: **Success** => S_c(Key,0) ∈ CST

If repeated enough time the **most frequent status is the expected one !**

Determination of more statuses

One triplet of clocks → 6 linear relations between the bits of the key

In order to execute the brute force step in a reasonable amount of time, 20 equations are required (at least)

The precedent step can be reproduced with the clocks (103,103,103)

→ only 3 more bits as the three other bits are already recovered

The NTW approach:

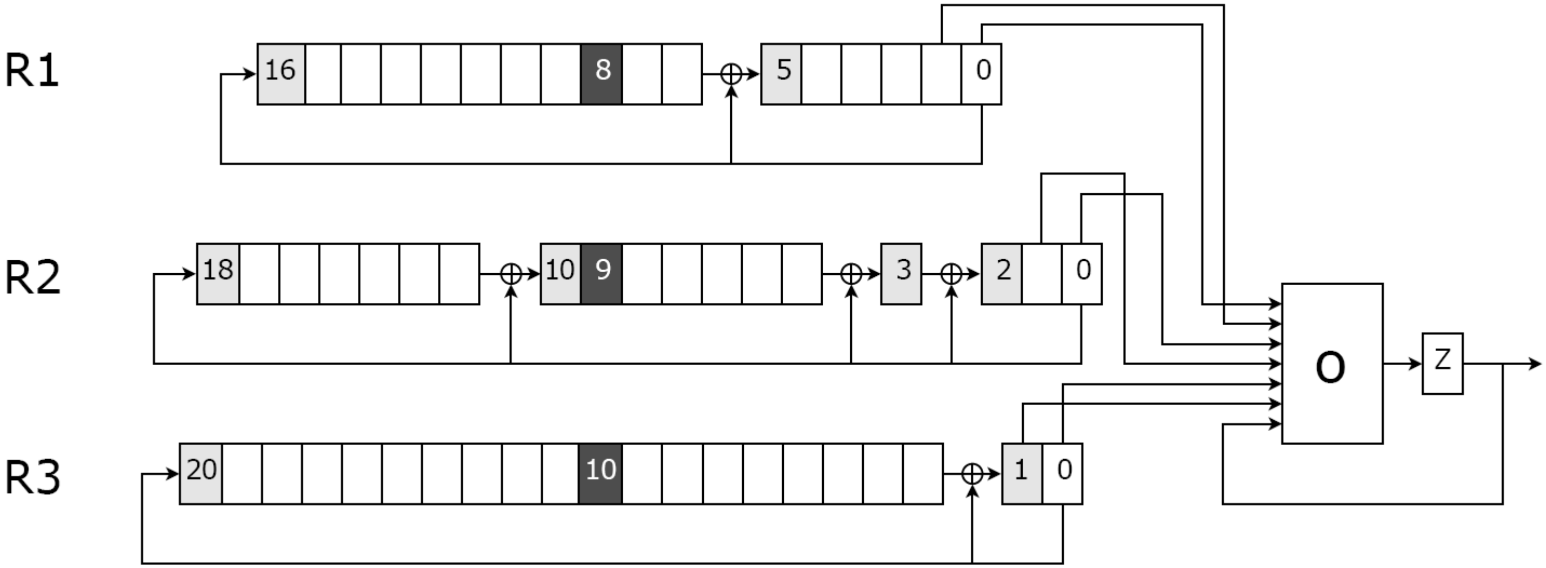
- Extend the attack to a range of 35 clocks for 19 bits of keystream
- Define a frequency table for each of the involved bits
- 108 equations are defined by these bits
- Select a solvable sub-system of equations
- Brute force the remaining bits

Determination of more statuses

One triplet of clocks → 6 linear relations between the bits of the key

In order to execute the brute force step in a reasonable amount of time,

2



The

The

- I
- Define a frequency table for each of the involved bits
- 108 equations are defined by these bits
- Select a solvable sub-system of equations
- Brute force the remaining bits

Determination of more statuses

One triplet of clocks → 6 linear relations between the bits of the key

In order to execute the brute force step in a reasonable amount of time, 20 equations are required (at least)

The precedent step can be reproduced with the clocks (103,103,103)

→ only 3 more bits as the three other bits are already recovered

The NTW approach:

- Extend the attack to a range of 35 clocks for 19 bits of keystream
- Define a frequency table for each of the involved bits
- 108 equations are defined by these bits
- Select a solvable sub-system of equations
- Brute force the remaining bits

Determination of more statuses

Our approach:

- Consider the entire status for a given range of len_c clocks
  - irrelevant candidates are discarded in the first step
  - Take into account all the “relevant” combinations of clocks for the first byte of the plaintext
  - $3(len_c + 1)$ equations are defined
- As in NTW we give a score to the candidates in each CST based on the probability that the targeted candidate is inside
  - refined probability model compared to the NTW attack
- Apply a time accuracy trade-off to remain efficient
- Even if not considered in the results, we obtain an ordered list of potential candidates based on their likeliness.

Theoretical and Experimental Results

Results based on Simulated Data

Details of the experiments:

- 200 DSC keys
- First IV randomly produced, the subsequent IVs incrementally
- Considering both C-Channel and B-Field
- Range of 12 clocks divided in 4 sub-ranges of 3 clocks
  - 39 equations
  - Discarding the two extreme bits reduces to 33 equations but increases significantly the success

Brute-force step:

- CPU SIMD-based implementation with a Core i7 (AVX) workstation
- $1 - 2^{-64} \approx 100\%$ probability of success
- Around 5 seconds for 25 bits

Results based on Simulated Data

Number of plaintext	4096	8192	16384	32768
10 equations (NTW)		2 %	30 %	96 %
9 equations (IS)	**35 %**	**85 %**	**98 %**	
20 equations (NTW)		0 %	2 %	78 %
21 equations (IS)	**16 %**	**73 %**	**97 %**	
30 equations (NTW)		0 %	1 %	48 %
33 equations (IS)	**6 %**	**55 %**	**95 %**	
40 equations (NTW)		0 %	0 %	11 %
39 equations (IS)	**2 %**	**33 %**	**84 %**	

Comparison of the success of the NTW attack and our attack against the C-Channel depending of the number of produced equations

European Commission

Results based on Simulated Data

Number of plaintext	8192	16384	32768	65536
10 equations (NTW)		2 %	30 %	92 %
9 equations (IS)	**19 %**	**69 %**	**94 %**	
20 equations (NTW)		0 %	2 %	65 %
21 equations (IS)	**10 %**	**57 %**	**90 %**	
30 equations (NTW)		0 %	0 %	28 %
33 equations (IS)	**3 %**	**36 %**	**82 %**	
40 equations (NTW)		0 %	0 %	4 %
39 equations (IS)	**1 %**	**21 %**	**66 %**	

Comparison of the success of the NTW attack and our attack against the B-Field depending of the number of produced equations

Extraction of Plaintext from Real Communications

Details of the experiments:

- Conducted against several phones from different brands
- Recording silence (1111..1111) in an anechoic chamber → well… no
- Pairing attack to know the plaintext with 100% accuracy
- 5 minutes of communication to collect 32K samples of B-Field

The accuracy of the "pure silence" ranges from 85 to 90%

- Surprisingly the attack was still successful
- The loss of accuracy can be compensated
  - by analysing more plaintext
  - by increasing the threshold N_T
  - the distribution of zeros is not uniform
- Simulation of communication for the B-Field for several degrees of inaccuracy

Results with a Reduced Accuracy

	32768 plaintexts				65536 plaintexts			
Accuracy	100%	95%	90%	85%	100%	95%	90%	85%
9 equations	96 %	92 %	71 %	55 %	100 %	100 %	100 %	92 %
21 equations	91 %	78 %	57 %	37 %	100 %	100 %	96 %	81 %
33 equations	85 %	65 %	42 %	21 %	99 %	98 %	87 %	70 %
39 equations	81 %	56 %	28 %	11 %	99 %	94 %	85 %	63 %

Comparison of the success of our attack (Top 50) against the B-Field depending of the number of produced equations for several levels of inaccuracy

Conclusion

- **In an ideal scenario, our improved known-plaintext attack can decrypt a communication with less than 3 minutes of communication intercepted with our SDR technic**

- **The attack is still feasible if the plaintext recovery is not perfect**

- **Our attack can be improved**
  - Some particularities of the output combiner are not used
  - Patterns in the bitstream generated by the voice codec can lead to a better prediction of the plaintext

➔ The DECT Stream Cipher 2 should sort out this issue. We hope our results could get translated in a wider adoption of DSC2

Stay in touch

JRC Science Hub: www.ec.europa.eu/jrc

Twitter: @EU_ScienceHub

LinkedIn: european-commission-joint-research-centre

YouTube: JRC Audiovisuals

Vimeo: Science@EC

European Commission