CHES 2015 Challenge

Adrian Thillard, Ryad Benadjila, Emmanuel Prouff, Guénaël Renault, Matthieu Rivain

CHES 2015 – Tuesday, September 15th, St-Malo, France

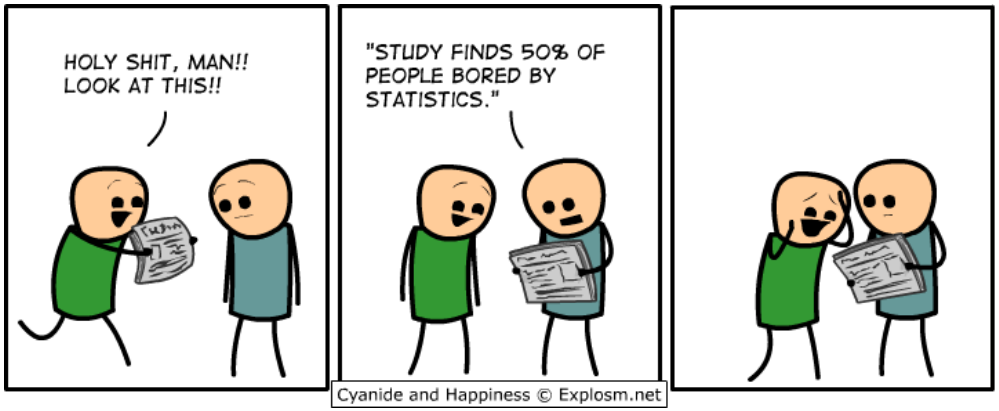
CHES Challenge : goal

- Challenge people on CHES topics
- Add fun to the conference

CHES Challenge : general principle

1. Download 4 challenges
2. Solve their problem to retrieve flags
3. Enter flags on our website to earn points
4. ???
5. PROFIT

Stats



- 250 registrations
- 44 retrieved at least one flag
- First to retrieve all the flags : 6 days - yobibe
  - Check his awesome writeup [1] ! !
- 8 players retrieved all the flags

1. `http://wiki.yobi.be/wiki/CHES2015_Writeup`

Winners (1/2)

1. hellman
2. yobibe (represented by Joppe BOS)
3. jybu (represented by François DASSANCE)
4. fox (represented by Ilya KIZHVATOV)

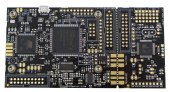
Winners (1/2)

1. hellman
2. yobibe (represented by Joppe BOS) 
3. jybu (represented by François DASSANCE)
4. fox (represented by Ilya KIZHVATOV)

Winners (1/2)

1. hellman
2. yobibe (represented by Joppe BOS) 
3. jybu (represented by François DASSANCE) 
4. fox (represented by Ilya KIZHVATOV)

Winners (1/2)

1. hellman
2. yobibe (represented by Joppe BOS) 
3. jybu (represented by François DASSANCE) 
4. fox (represented by Ilya KIZHVATOV) 

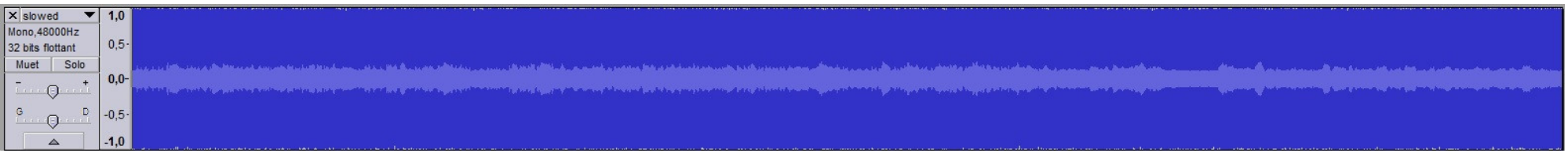
Winners (2/2)

5. c23 (represented by Cyril ROSCIAN)
6. Seeluna (Céline THUILLET)
7. barbapapa (represented by Julien FRANCQ)
8. OverTime (represented by Alberto BATTISTELLO)
9. dummy (represented by Peter SHWABE)
10. marsob

CHES Challenge : description

1. WAV file : signal analysis, SCA
2. JPG file : fun (stegano, chess, googling)
3. C file : factorisation, primes collision, SCA, fault attacks
4. PNG file : pattern matching, emulation, padding oracle, whitebox

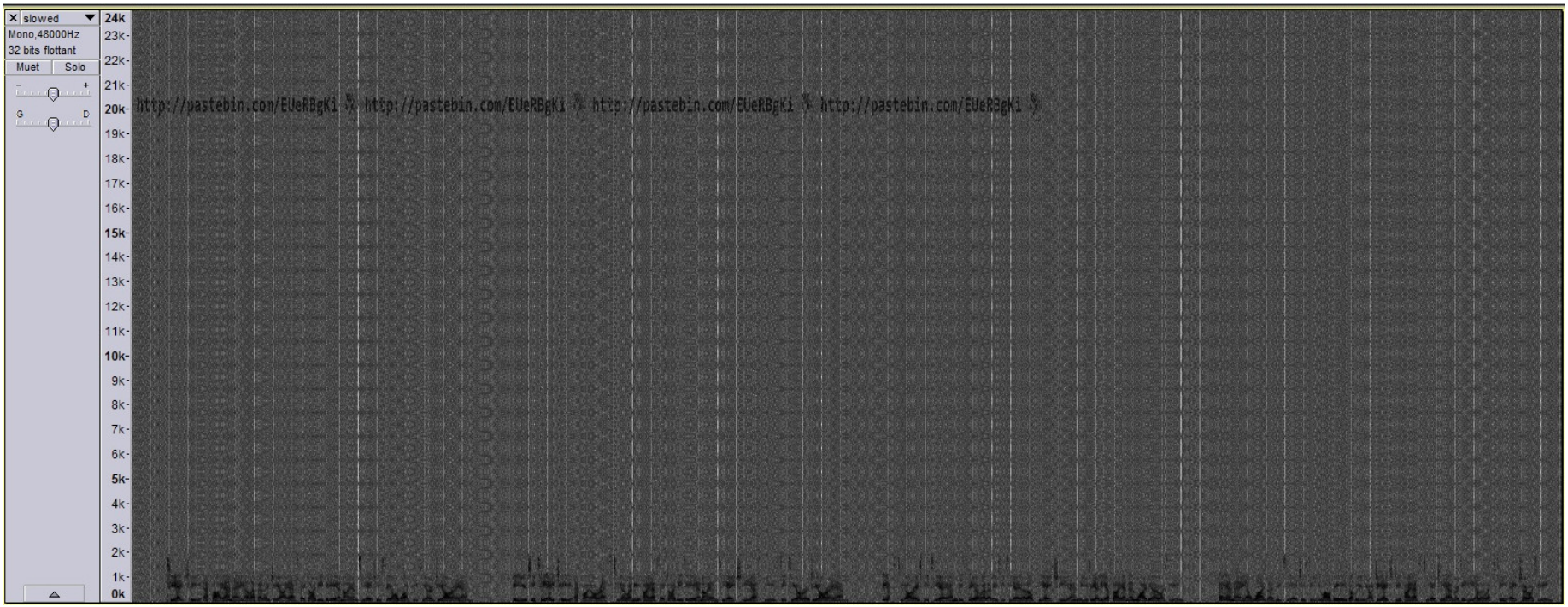
Challenge 1- WAV file



Challenge 1- WAV file : First flag

- Quicken the file $\Longrightarrow$ voice reading letters
- Letters form sentences $\Longrightarrow$ solving recipe

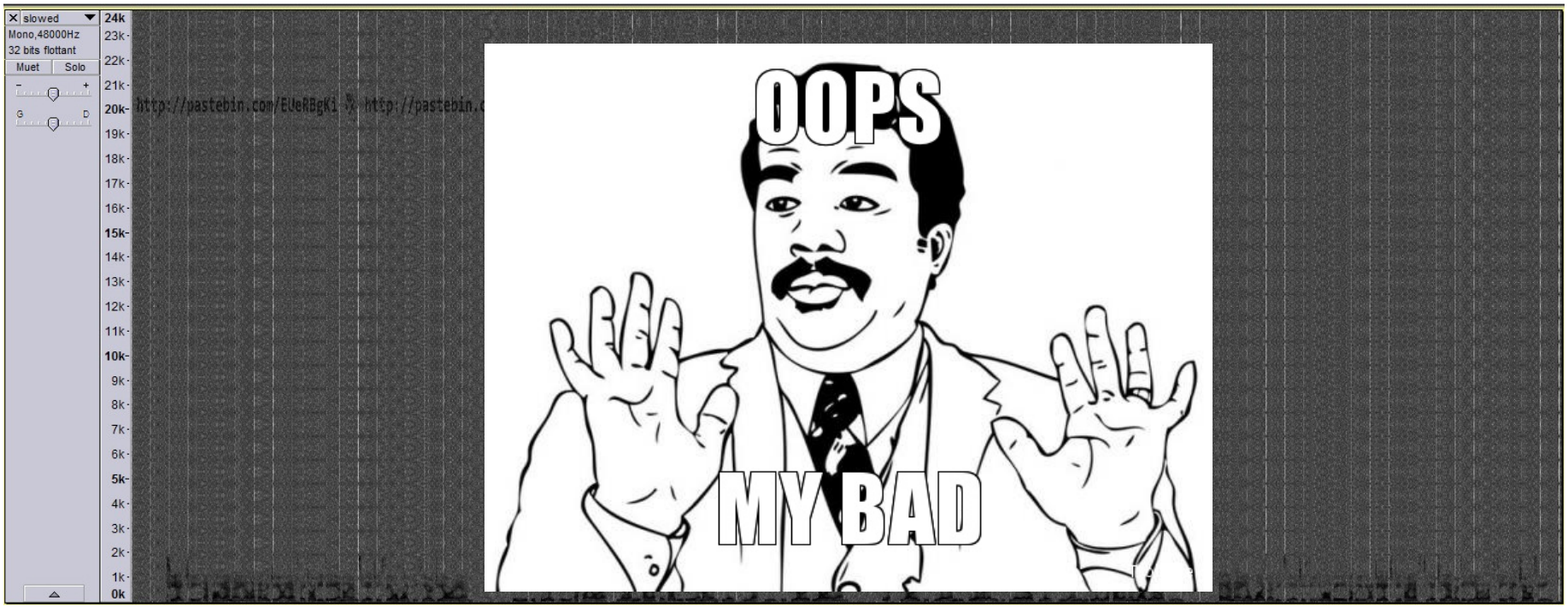
Challenge 1- WAV file : First flag is in the spectrogram



Go on pastebin $\Longrightarrow$ first flag and plaintexts[1]

1. Note to self : do not screw with the plaintexts

Challenge 1- WAV file : First flag is in the spectrogram

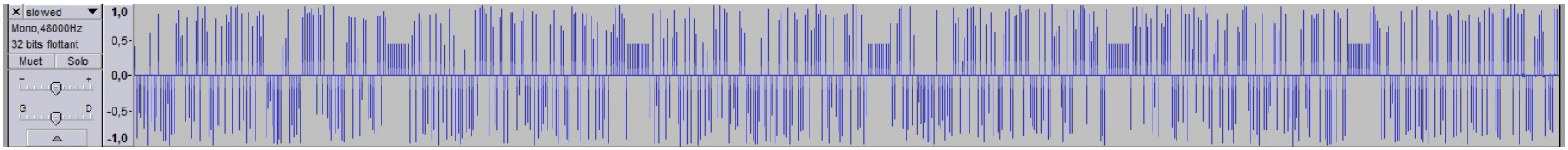


Go on pastebin $\Longrightarrow$ first flag and plaintexts[1]

1. Note to self : do not screw with the plaintexts

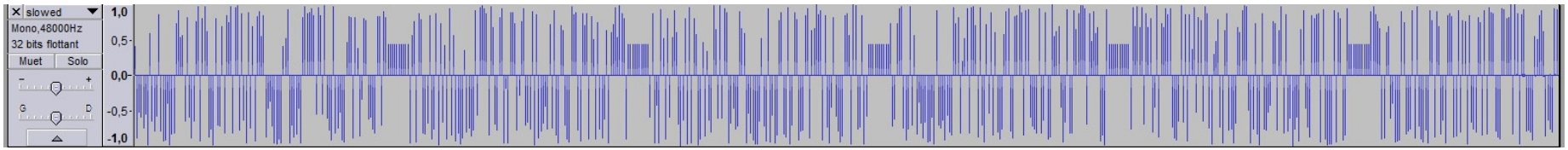
Challenge 1- WAV file : Second flag : Getting the curves

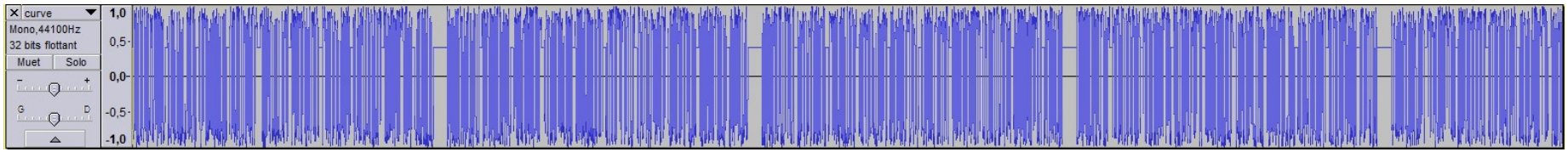
Recipe instructed to extract needles



Challenge 1- WAV file : Second flag : Getting the curves

Recipe instructed to extract needles

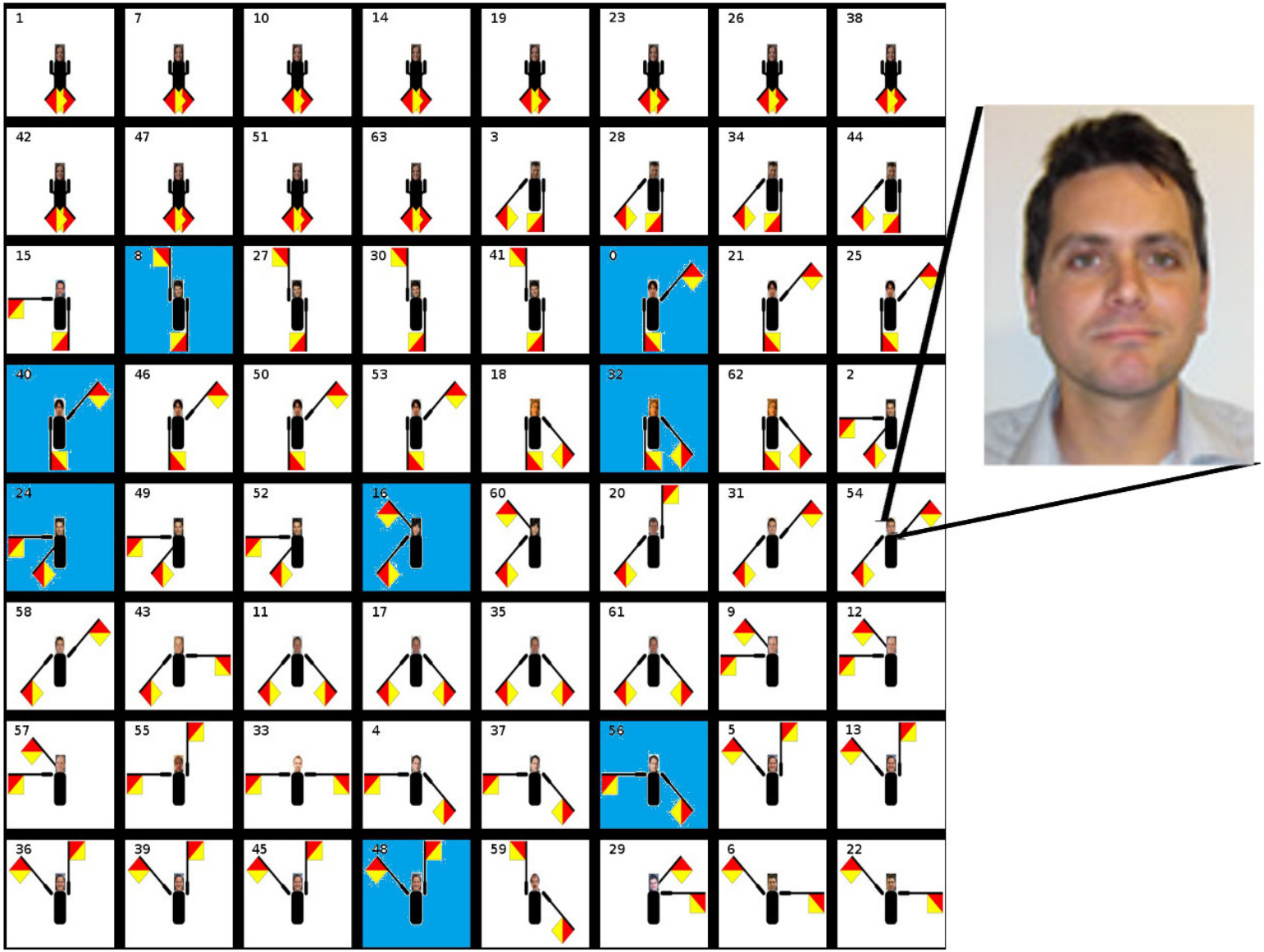




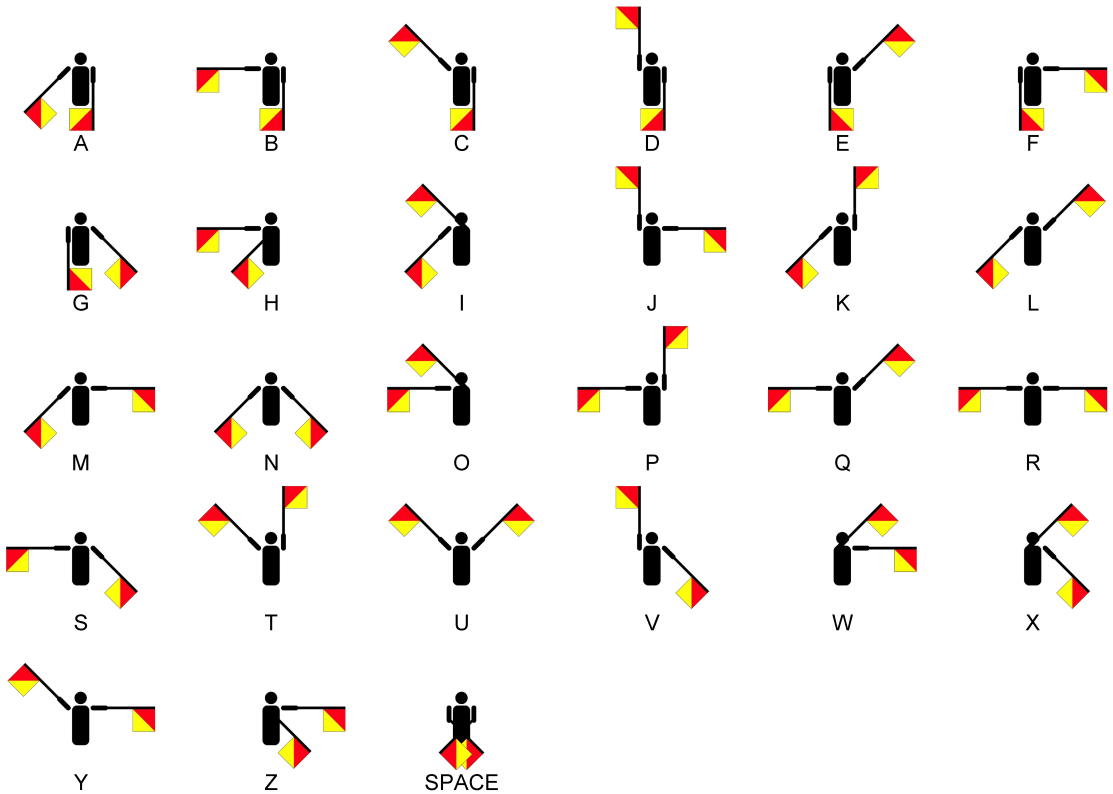
Challenge 1- WAV file : Second flag

- CPA HW (Mayer-Sommer (CHES00), Brier et al. (CHES04)) $\Longrightarrow$ Secret Key
- Secret Key $\Longrightarrow$ flag

Challenge 2- JPG file



Challenge 2- JPG file : First flag



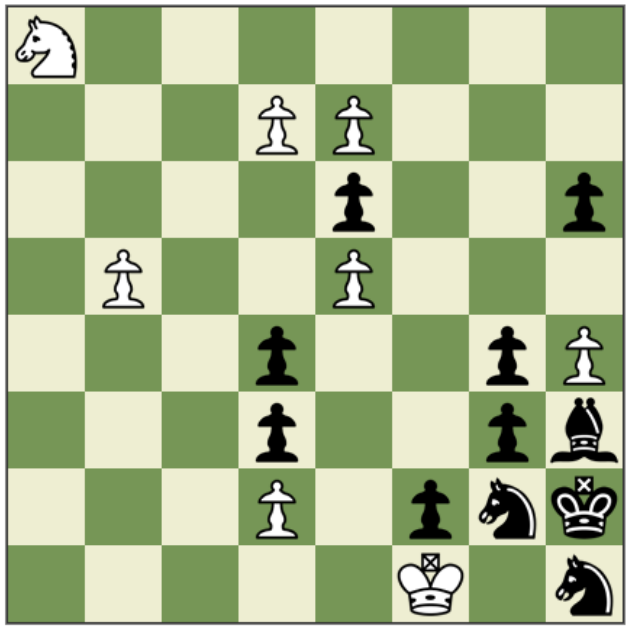
credit : Denelson83

Challenge 2- JPG file : First flag

- Order the cells according to their numbers
- Blue $\Longrightarrow$ STEGHIDE, phrase about helped mate
- Use STEGHIDE on jpg with password = previous phrase
- Get Gabor.txt $\Longrightarrow$ first flag

Challenge 2- JPG file : Second flag

In text file : FEN notation



Challenge 2- JPG file : Second flag, path 1 : Solve it !

Challenge 2- JPG file : Second flag, path 2 : Google it !

Cseh.jpg + Gabor.txt $\Longrightarrow$ Gabor Cseh

Cseh, Gábor

The Problemist 1997

1st Prize

N7/3PP2p/4p3/1P2P3/3p2pP/3p2pb/3P1

h#10 8+11

Solution:

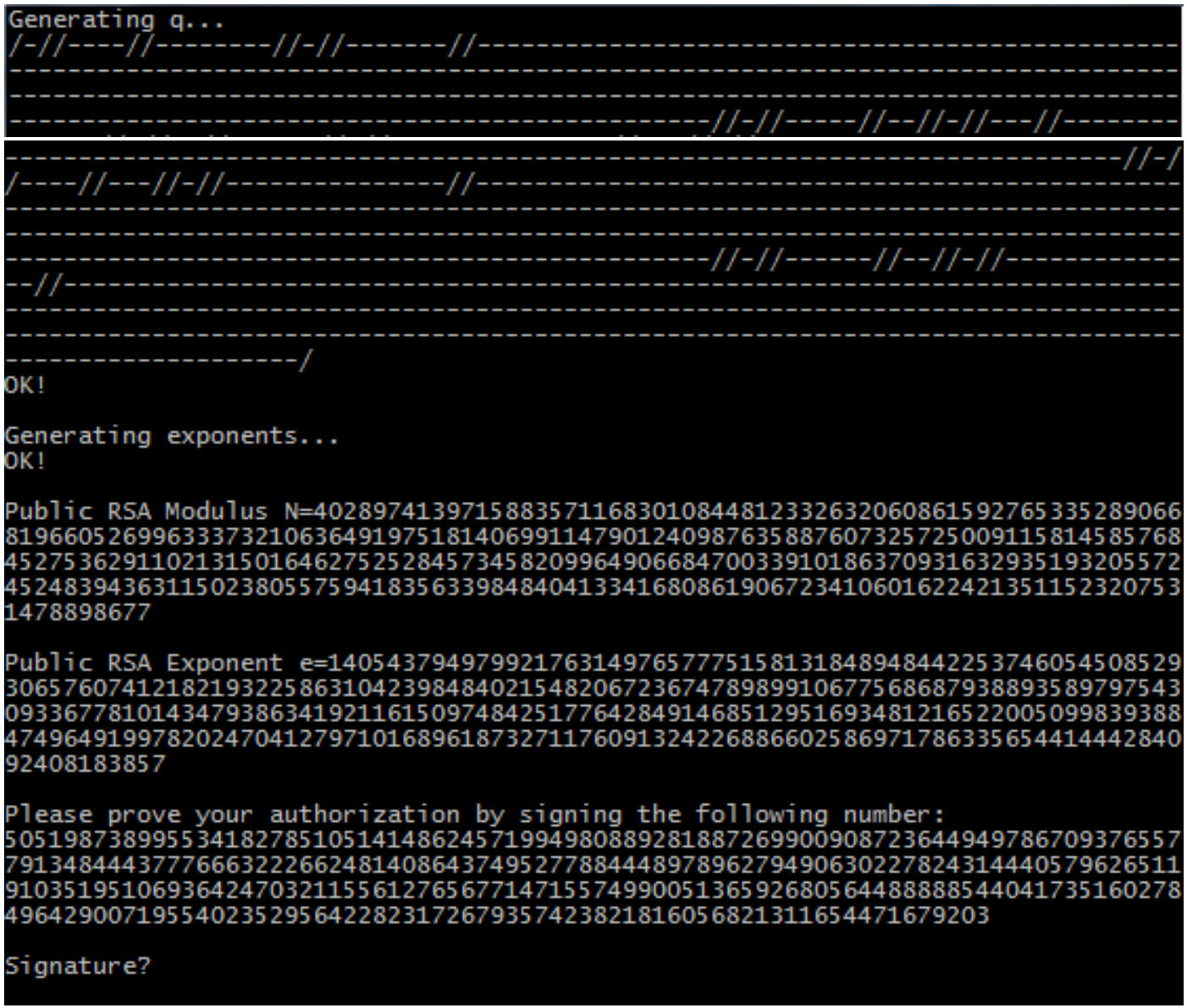
1.h6 d8Q 2.h5 Qd5
3.S:h4+ Qg2+ 4.S:g2 e8B
5.h4 Bc6 6.Sf4+ Bg2
7.Sd5 b6 8.Sc7 b7
9.S:a8 b:a8Q 10.B:g2+ Q:g2#

References (1)

Challenge 2- JPG file : Second flag

- Encode each move using grid numbers (eg. G2-H4= 14,31)
- Secret Key $\Longrightarrow$ flag

Challenge 3- C file : Behavior



Challenge 3- C file : Behavior

Wrong signature :

```
Signature?
kamoulox
Deciphering AES ciphertext...
10
9
8
7
6
5
4
3
WARNING!!!!!!!!! WRONG AUTHENTICATION!!UNAUTHORIZED ACCESS!
2
1
Plaintext=a46bc574cda56ac97e7973a12d5bfdd2
```

Correct signature :

- Correct plaintext ? ? ?

Challenge 3- C file : First flag, path 1 : side-channel

Prime generation by trial divisions

- Generate random
- "-" $\Longrightarrow$ not divisible
- "/" $\Longrightarrow$ divisible $\Longrightarrow$ random+1

Challenge 3- C file : First flag, path 1 : side-channel

Ideal application of Finke et al. (CHES09) :

- Get a lot of modular equations involving the prime
- Solve them using CRT
- Factorize N

Challenge 3- C file : First flag, path 2 : prime collision

Only 100 different primes can be generated by the server

- Build $\{N_1, N_2, \cdots\}$
- Compute $\gcd(N, N_1)$, $\gcd(N, N_2) \cdots$
- Factorize N when gcd $\neq 1$

Challenge 3- C file : First flag

```
Public RSA Modulus N=557046461951618065697121293617150244352268554520986511224937552001497910951980760019207189278232276873826539990856921333078895682225378296083945174990155754504168420629391446992695730208712200683127938786203538898623223315215072299527296354558611631927024971970574227830594411303833764208141102423303066067517

Public RSA Exponent e=325501310250801167192705350467048355613383406747909306639372398068528267955730707367342301121423194603738915656831855065231282314817280444777443793254164722581316476858589232455061610683555659583355848629184769257451544674659192315730974098728970307337965434976120697179310834791029552105188244367766257580061

Please prove your authorization by signing the following number:
369353181134346100135393034081877740745777771368173096792378612239670779306102208719020770303950336154793637206735426571126202279999143580530025979719332231321785541087211316979965074944475848883422118088374823783648888377152378322738653391083474602587392499895022361450055631935000352712684859595375518137802 8

Signature?
745208204114328303871058989378378819655388867468914244919950335298043479996333005909411342660992589351801310728088366079905623607915320442708043660769691001766007244059474569367346441647297981276054852854226808797528254926640346309013870714921761942990475330302203842624308424701795931356602642400584915726 44
Deciphering AES ciphertext...
10
9
8
7
6
5
4
3
2
1
AuthentiAuthentification complete!Plaintext=2516ab30e01bc44a3e9f710495f6dc3e
```

- First ciphertext only 4 blocks
- Use server as decryption oracle $\Longrightarrow$ flag

Challenge 3- C file : Second flag

- Second ciphertext is a several hundreds of MB picture
- Too long to use previous method[1] ($\geq$ month)

1 : Note to self : do not screw the server implementation, it could be DoS'ed otherwise.

Challenge 3- C file : Second flag

- Second ciphertext is a several hundreds of MB picture
- Too long to use previous method[1] ($\geq$ month)

1 : Note to self : do not screw the server implementation, it could be DoS'ed otherwise.

Challenge 3- C file : Second flag, path 1 : clever server heckler

Ask for decryption of random blocks of the picture

- Blank space $\Longrightarrow$ change area
- Black zone $\Longrightarrow$ useful info $\Longrightarrow$ decrypt foreign blocks

Decryption of useful parts $\Longrightarrow$ flag

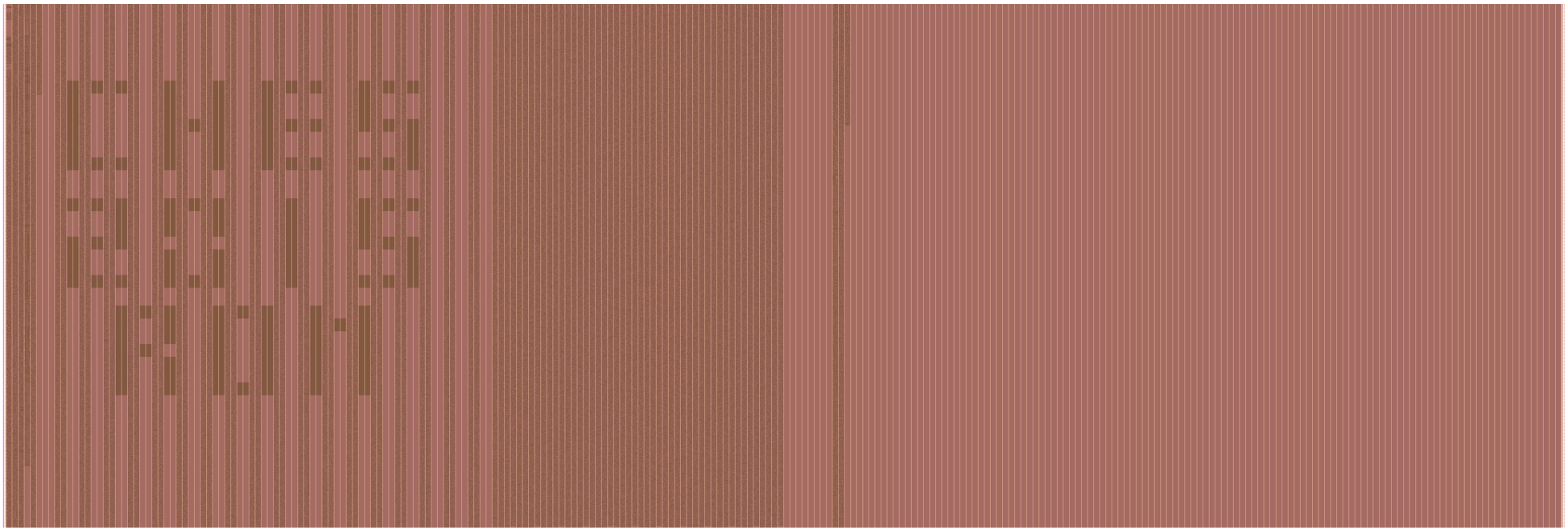
Challenge 3- C file : Second flag, path 2 : fault attack

Ask for two decryptions of the same block

- Answer wrongly $\Longrightarrow$ error in the 2nd to last round $\Longrightarrow$ C^*
- Answer correctly $\Longrightarrow$ C

Piret and Quisquater (CHES03) on AES decryption :
$(C^*, C) \Longrightarrow$ secret key $\Longrightarrow$ flag

Challenge 4- PNG file



Challenge 4- PNG file : First flag

Pattern matching :

- On cell $\Longrightarrow$ bit 1
- Off cell $\Longrightarrow$ bit 0

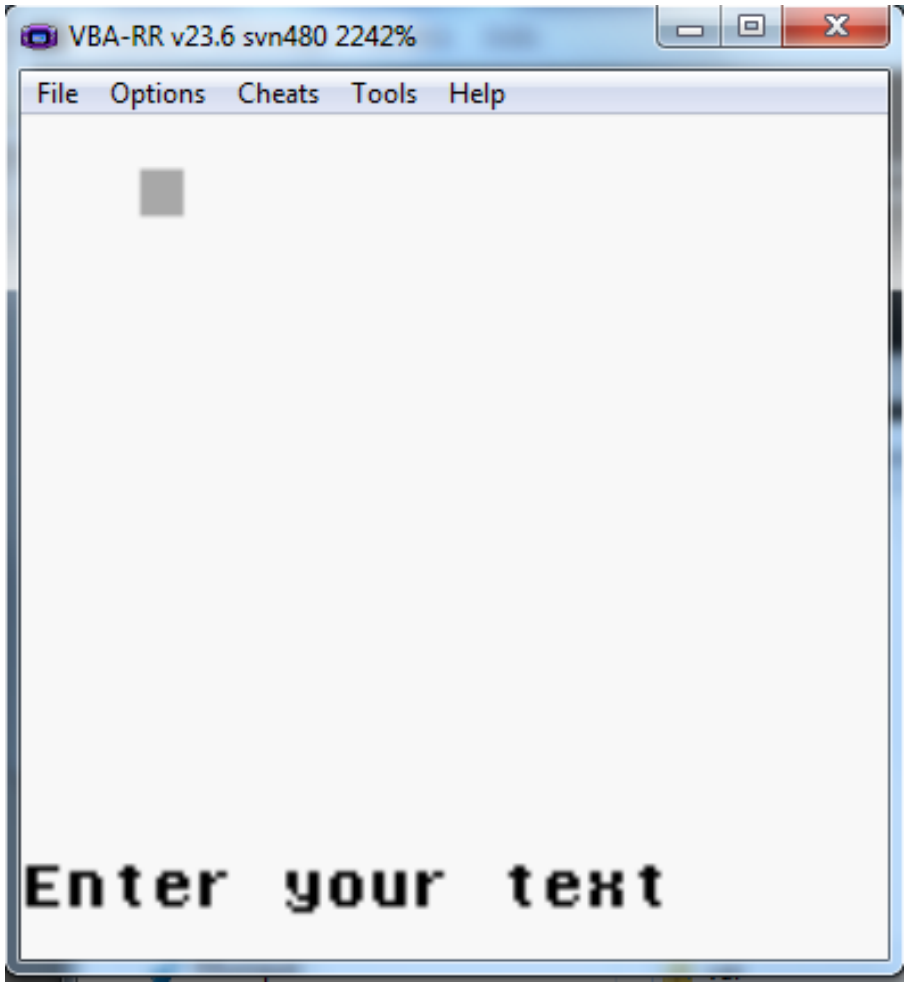
Challenge 4- PNG file : First flag, path 1 : static analysis

- Look at strings
- Get flag (one of the only strings that is not obfuscated)

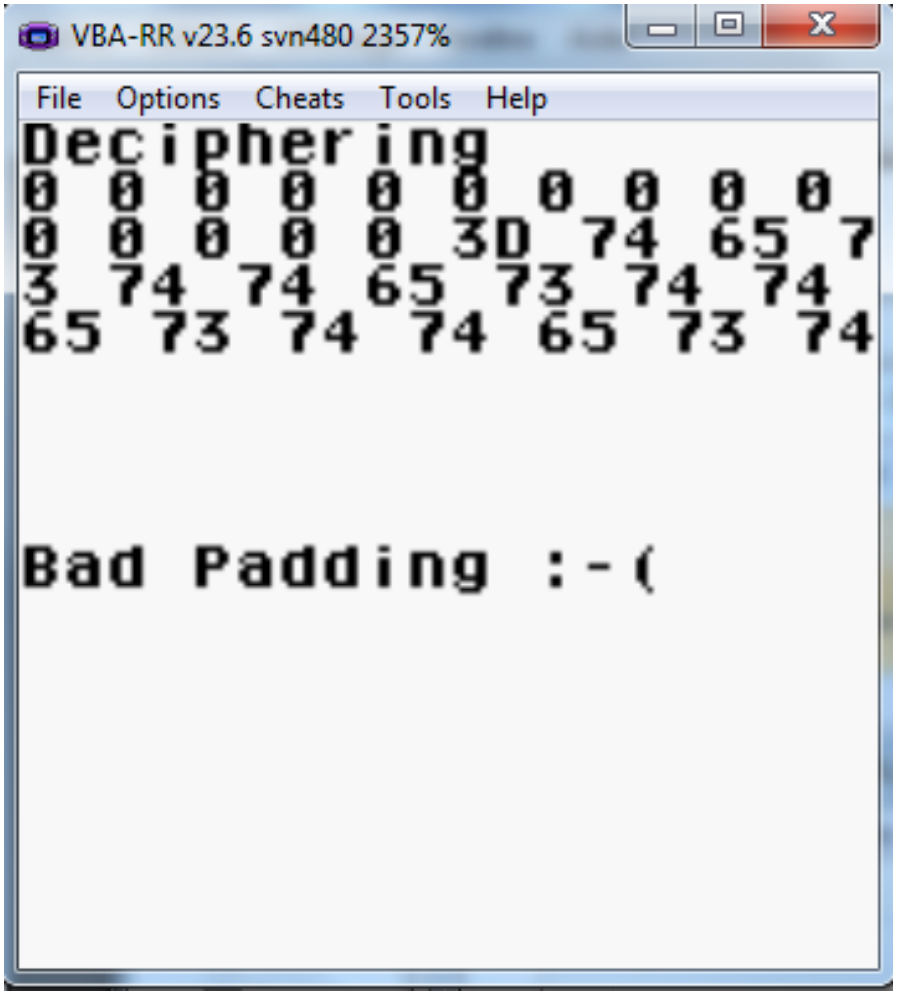
Challenge 4- PNG file : First flag, path 2 : emulation

- Command `file` $\Longrightarrow$ GameBoy ROM
- Launch a GB emulator $\Longrightarrow$ flag

Challenge 4- PNG file : Second flag, path 1 : emulation



Challenge 4- PNG file : Second flag, path 1 : emulation

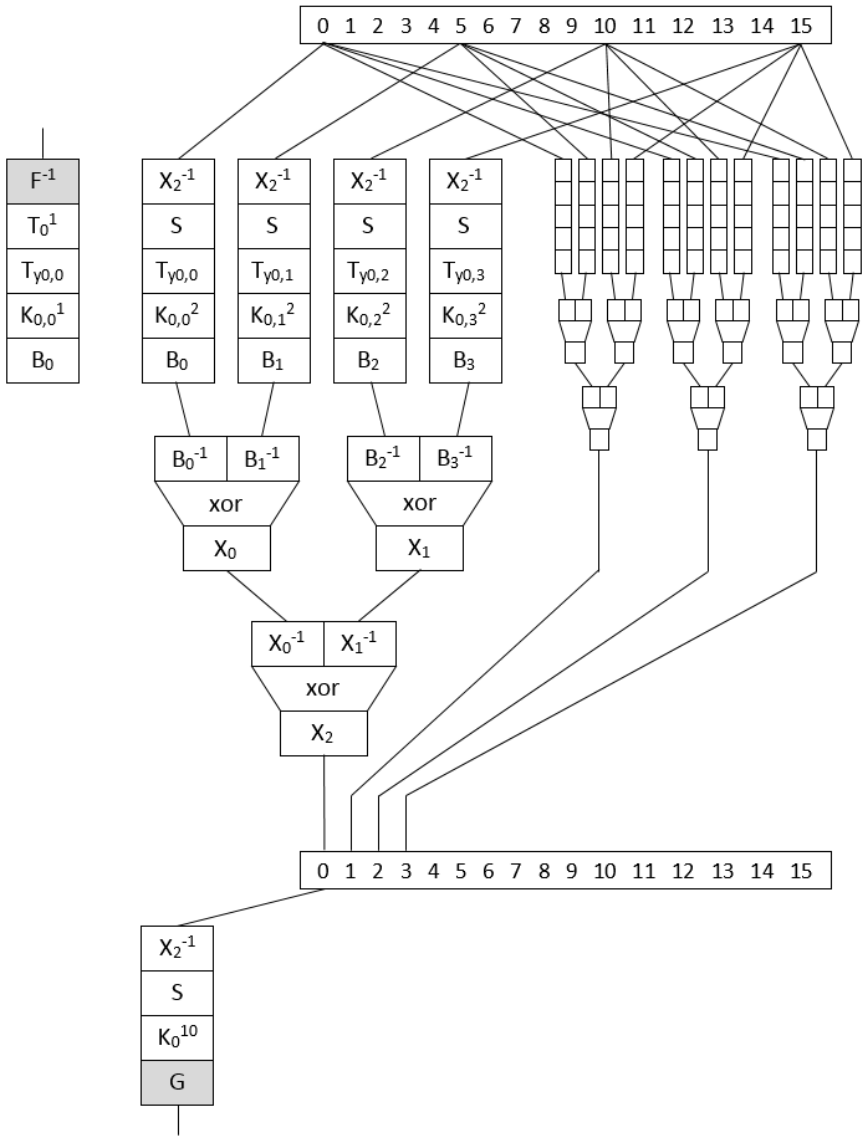


Challenge 4- PNG file : Second flag, path 1 : emulation

- ROM is a padding oracle on AES-CBC
- Vaudenay's attack (EUROCRYPT02) $\Longrightarrow$ decryption [2] oracle
- Script an attack (Lua scripting with Visual Boy Advance or emulator patching)
- Decrypt ciphertext $\Longrightarrow$ flag

2. Except we have encryption here : same attack applies !

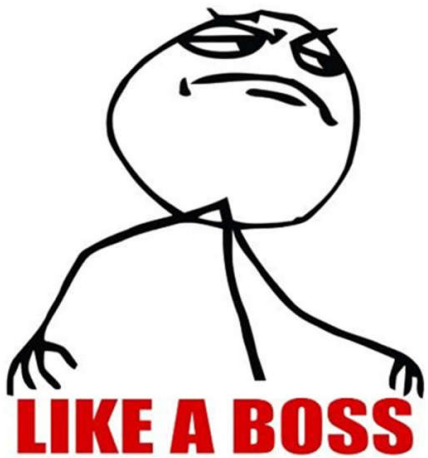
Challenge 4- PNG file : First flag, path 2 : WB pwning



credit : yobibe

Challenge 4- PNG file : First flag, path 2 : WB pwning

- Reverse soft and GB architecture (memory banks, etc.)
- Break whitebox
- Secret key $\Longrightarrow$ flag



Acknowledgments

We'd like to thank the following persons for their help in the conception and testing : Aurélie Bauer, Sonia Belaïd, Guillaume Bouffard, Jean-Christophe Delaunay, Thomas Fuhr, Emilien Girault, Pierre-Michel Ricordel, Joana Treger-Marim, Philippe Valembois, Eloi Vanderbeken, and all the persons on this obscure GB-ROM dev IRC channel that insisted half an hour on the fact that implementing a crypto algorithm on the GameBoy was useless. Martin also insisted for special thanks to Jacquie & Michel.

Call for challenge

- There will be a challenge next year
- More information coming soon

We want you ! ! !