



# Passive and Active Leakage of Secret Data from Non Networked Computers

Vincent Calmette-Vallet

Stéphane de Royer-Dupré

*Eric Filiol (speaker)* efiliol@esat.terre.defense.gouv.fr

Guy Le Bouter

**French Army Signals Academy (ESAT)/ESIEA Laval**
**Virology and Cryptology Lab.**
**France**


esiea

# INTRODUCTION

- Users consider non networked computers as naturally secure and protected against data theft.
- Danger comes from the network:
  - LAN…
  - … and of course Internet.
- Right or wrong idea?
- Let us try to explain why it is a wrong idea!


esiea

# INTRODUCTION

- A (isolated) computer is quite never really disconnected from the outside:
  - Need for data exchange (von Neumann model).
- A simple USB key or a CDROM will indirectly connect such a computer with other ones.
- Infecting a computer is unfortunately still really easy:
  - Write an efficient malware!
  - Just use social engineering!
- Remaining undetected is unfortunately still far easier for targeted attacks:
  - Rootkits, sophisticated malware…



Black Hat Briefings


esiea

# INTRODUCTION

**Consider an infected computer.**

- How to make secret data evade while
  - no network connection is available,
  - or network connection is not usable (IPSec network).

**The solution is:**

- Use a natural covert channel…
- … or create it!


# INTRODUCTION

esiea

- Our approach:
  - We are going to explore some possibilities to solve this problem operationally.
  - Let us just suppose (without loss of generality) we want to make a 20-byte password evade from a computer.
  - This data is wiretaped by a keylogger or an equivalent malware.
- Can be applied more generally to files or any other data.


# AGENDA

- Introduction.
- The concept of covert channel.
- Existing (known) threats.
- Our first operational technique
  - The Windows jingle attack
- Our other attacks.
  - Visual covert channel.
  - Others…
- Protection, future work and conclusion.


esiea

# The concept of covert channels


esiea

# What is a covert channel?

Definition of the US DoD (1985) :

- Communication channel B which borrows part of the bandwidth of an existing communication channel A.
- Enables to transmit information without the knowledge/permission of the legitimate owner of channel A and/or of the data transmitted.


esiea

# Covert Channels Examples

- **Cryptography:**
  - Timing attack (Kocher – 1996).
  - Power analysis (Kocher – Jaffe – Jun – 1999)
  - BPA (Seifert et al. – 2007).
- **Virtual machine detection**
  - (Lauradoux – Eicar 2008).
- **Malware-based data leakage over IpSec channels**
  - (Filiol et al. – ECIW 2008).


esiea

# Covert channels #2

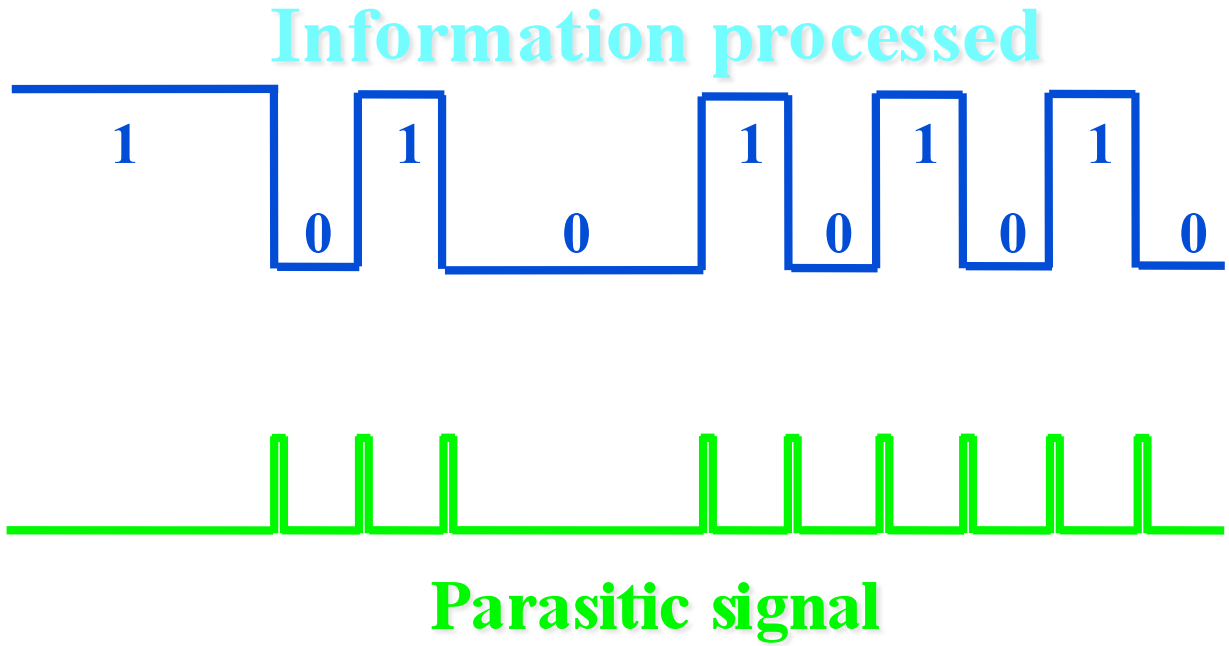
- Most of the known covert channels just exploit natural, physical or logical effects/properties of
  - *Hardware (e.g. power analysis).*
  - *Programs (e.g. timing attacks).*
  - *Protocols (e.g. IPSec).*
- It is possible to create new covert channels by subverting some normal features/properties.
- Quite everything in a computer can be turned into a covert channel!
- The bad news: a simple malware can do it!


esiea

# Existing (more or less known) Threats

# TEMPEST

- ***TEMP****orary* ***E****manation and* ***S****purious* ***T****ransmission*



# TEMPEST #2

- It is a physical effect (cannot be avoided).
- Different kind of propagation:
  - Aerial propagation (up to 150 meters).
  - Propagation by conduction (150 meters < ).
  - Pairing propagation (? meters).
- The parasitic signal frequency depends on
  - the hardware,
  - the velocity of the electronic commutation (signal of information processed).

# TEMPEST #3

- Parasitic signals can be remotely exploited very easily.
  - From a CRT monitor (van Eyck – 1985).
  - From a CRT monitor (Anderson & Kuhn – 1998).
  - From a LCD monitor (Kuhn – 2004).
  - Power Line Carrier (PLC)-like techniques.
- Any electrical/electronic device (including cables) is bound to emit such parasitic signals.
- With a lot of mathematics and physics, many things can be done with EM parasitic signals:
  - Classified information most of the times whichever country.


esiea

# TEMPEST #4

- Possible protections:
  - Faraday cage (very very expansive).
  - Tempest-proof hardware (very expansive).
    - Armoured hardware and cables.
  - Installation security rules (many constraints for the user).
- BUT all those protections can be bypassed by trapping or tampering the hardware:
  - The Moscow bug http://www.time.com/time/magazine/article/0,9171,958127-3,00.html
  - ... or use malware (officialy no known cases)!
- Mathematics and signal processing are progressing.



# Existing (more or less known) Threats

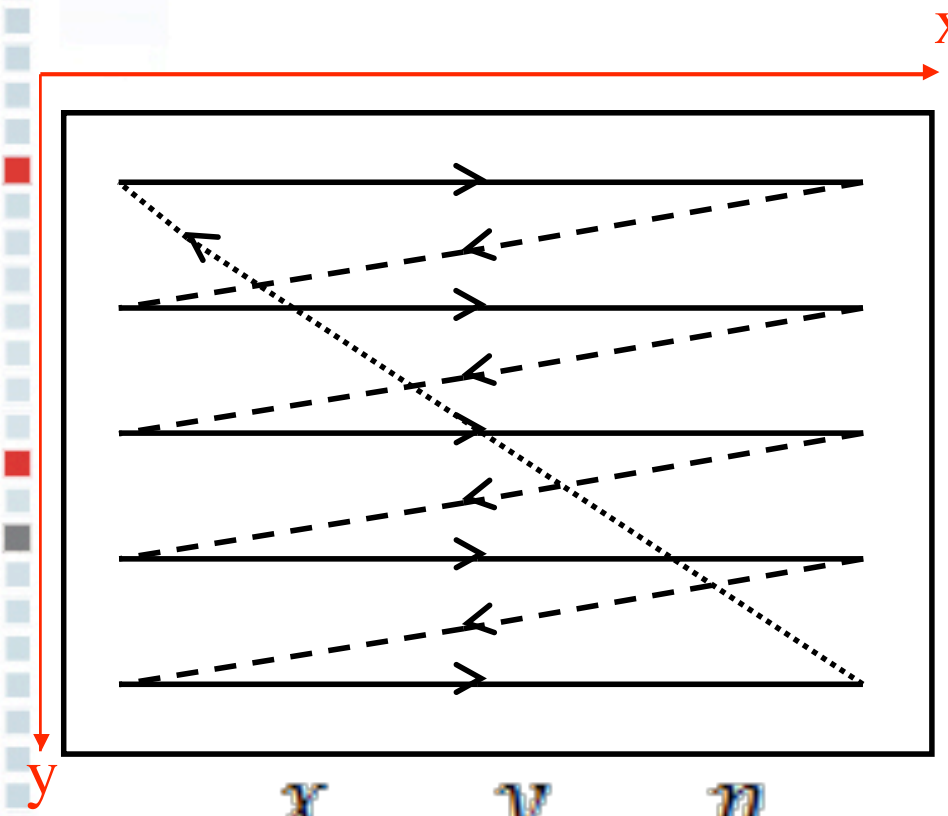
## Emitting an audio wave from a CRT monitor (Anderson & Kuhn's attack)


esiea

# Principles of the attack

- Implemented by Erick Thiele and inspired by Kuhn & Anderson's work in 1998.
- Tools available at http://www.erikyyy.de/tempest/
- Generate and broadcast an approximate (AM amplitude modulation) radio wave from a monitor display.
- Can be intercepted by a basic AM radio set.
- The intercepted signal is then processed to recover the secret data.

# Principle of the attack #2




Information writing

Horizontal flyback

Vertical flyback

$$t = \frac{x}{fp} + \frac{y}{fh} + \frac{n}{fv}$$

$f_p$: pixel clock frequency.
$f_h$: horizontal freq (line).
$f_v$: vertical freq (image).
n: frame counter

# Principle of the attack #3

- Let s(t) be the (radio) signal to be emitted.

$$0 < \frac{255}{2}\cos(2\pi.fc.t).[1+\cos(2\pi.ft.t)] < 255$$

Carrier freq. Audio tone freq.

Pixel display intensity is modulated according to a 256-grey scale

# Principle of the attack #4

- Screening (dithering) process:
  - Grey value V coded as a byte.
  - V = 255/2 + s(t) + R.
  - Choose arbitrary (unused) $f_c$ and $f_t$ freq (2 MHz and 300 Hz resp.).
- Theoretical transmission rate of 50 bits/sec.
- Secret data are then coded with FSK (0 and 1 represented by tone patterns).

# Principle of the attack #5

- Record the AM broadcast signal.
- Digitalization step.
- Symbol detection step.
- Synchronization step.
- Decoding step.
- Reasonably good interception at several hundred meters.
- Very simple technique and very limited hardware required (less than 100 $)!

# Principle of the attack #6

- Thiele's implementation (Tempest for Eliza) http://www.erikyyy.de/tempest/
- Signal can be intercepted tens of meters far from the target computer.
- Limitations: at emission the screen is scrambled.
  - Just wait for the user to be absent (suitable time period, keyboard inactivity…).
  - Thiele's demo…
  - Very powerful for a small amount of data (e.g. password).

Black Hat Briefings


esiea

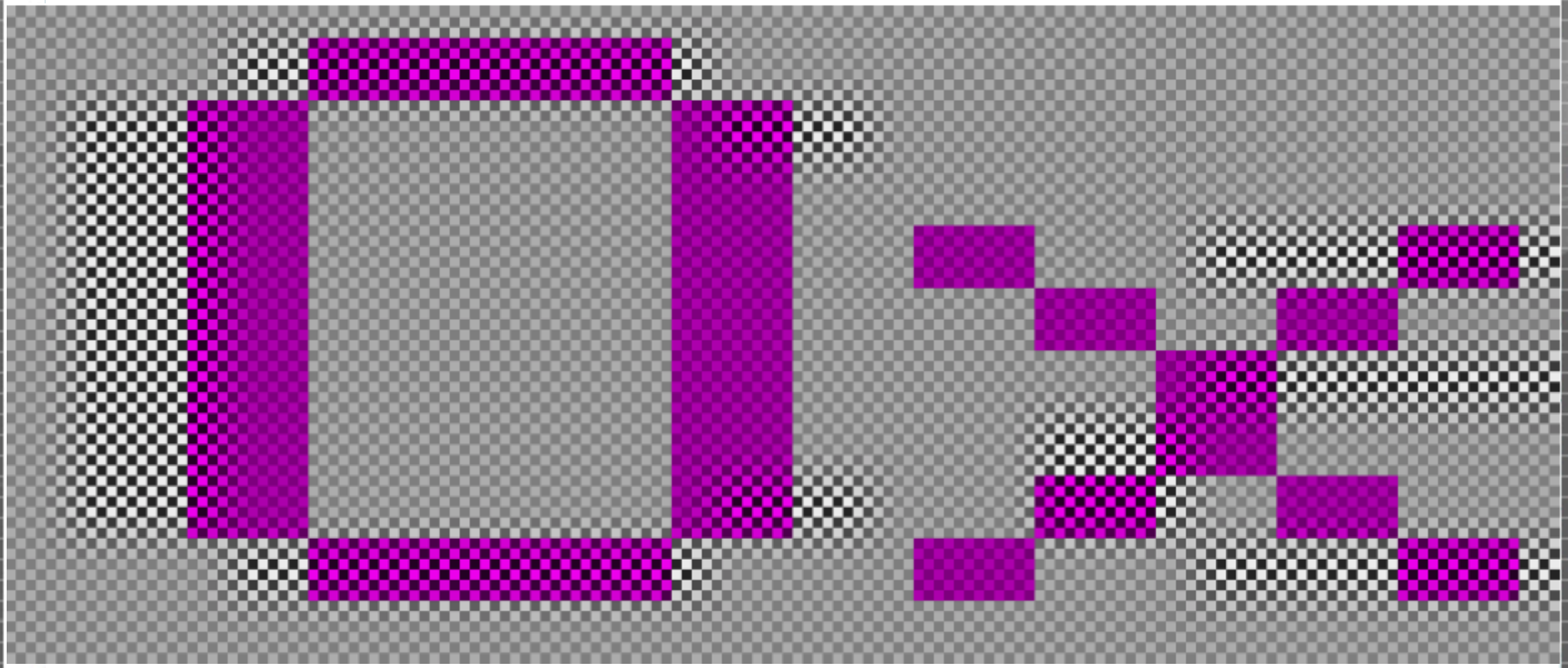
# Existing (more or less known) Threats

## Dither Patterns Attacks (Anderson & Kuhn's attack #2)

# Principle of the attack

- Human eye is less sensitive to high than to low spatial frequencies.
- Halftoning (dithering) plays with that perception difference to increase the number of colours shades available.
  - User cannot distinguish between a medium grey and a chequered halftoning pattern of grey/white pixels.
  - On the attacker's side the high frequency black/white dither pattern creates the strongest possible signal.
- Spread-spectrum techniques and other optimizations possible.

# Dithering or halftoning technique

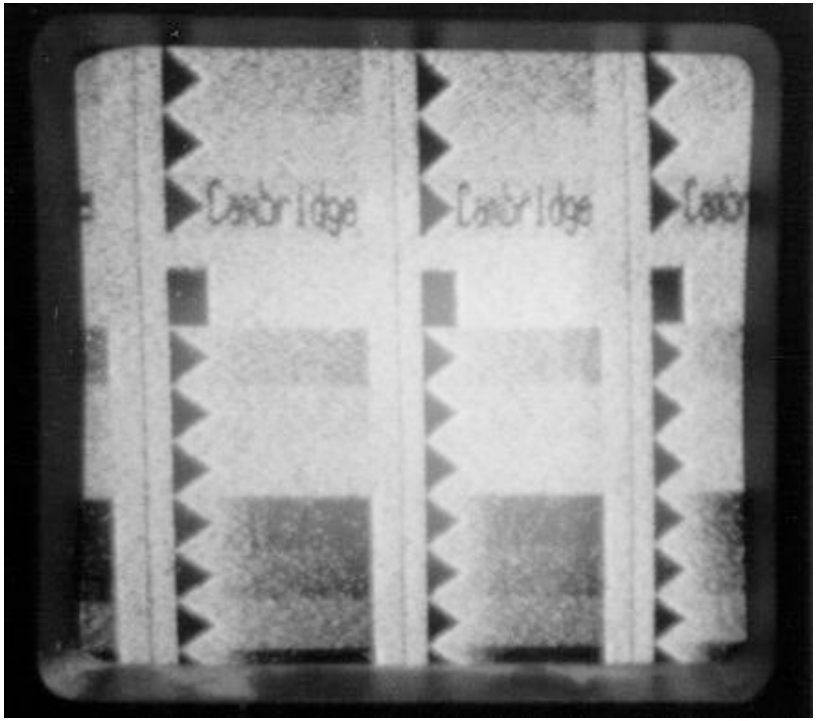


**Target screen**

**Magnified pixel around the letters « Ox » that radiate as « Ca »**

# Dithering or halftoning technique #2


Target screen



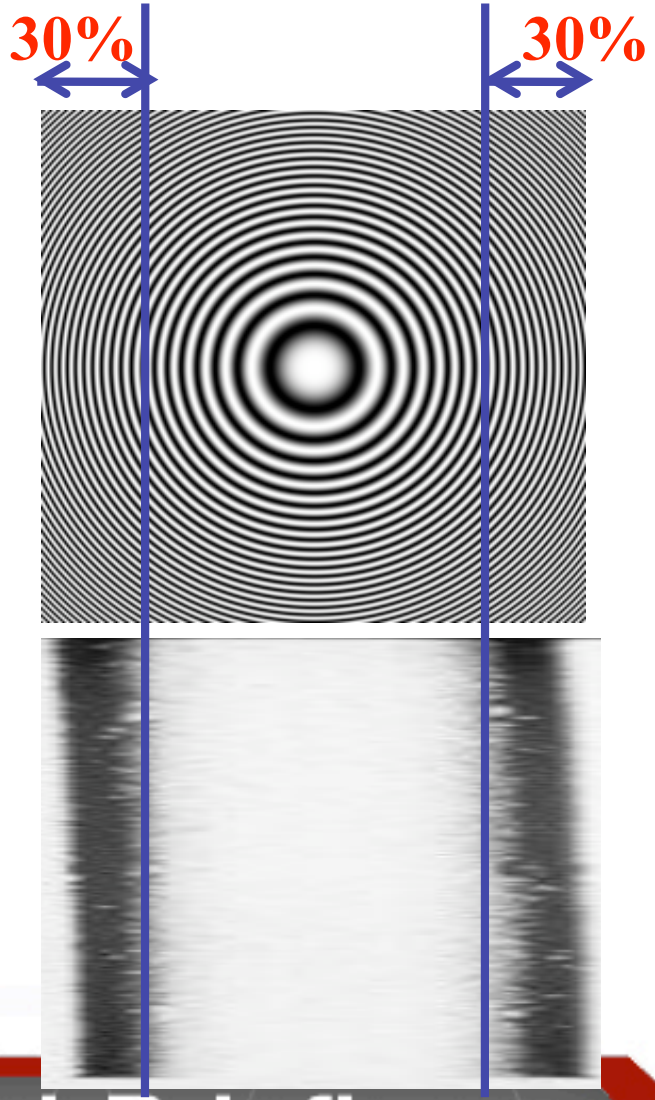
Attacker's screen


esiea

# Existing (more or less known) Threats

## Reversing Tempest Font Protection (Anderson & Kuhn's protection)

# Anserson & Kuhn's TEMPEST Fonts




Target screen

$\cos(x^2 + y^2)$

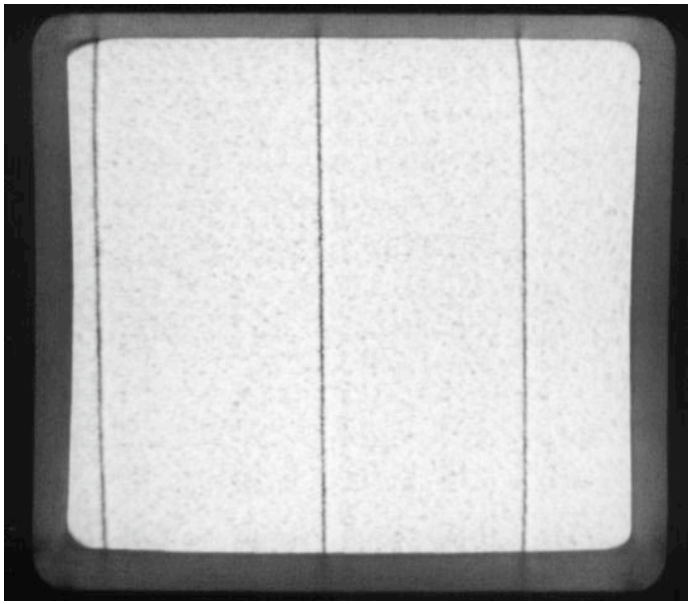
Attackers's screen

# Anserson & Kuhn's TEMPEST Fonts #2

**Design filtered (horizontal freq.) fonts.**

Attacker's screen

Normal fonts



TEMPEST fonts

# Anserson & Kuhn's TEMPEST Fonts #3

- Rather old results (1998).
- New significant progresses made in interception techniques.
- TEMPEST fonts are no longer a protection.
- New results to intercept Kuhn's TEMPEST fonts:
  - (Tanaka & al., 2004; Tanaka, 2008).
  - Use of Gaussian filters.
  - Recovering of 80 % of the TEMPEST fonts.
  - Proposed improved TEMPEST fonts.

# Reversing the principles

- Instead of filtering (LPF) the upper 30 % horizontal frequency, filter the lower part (HPF) and process the signal to make it easier to intercept.
- Attacker will intercept it more easily.
- A malware can substitute existing fonts with « radiating » fonts.
- Production of those fonts under current development.
- « Radiating » fonts are expected to be far less blurry that TEMPEST fonts.
- To be continued…


esiea

# Existing (more or less known) Threats

## Podslurping



# Podslurping

- The malware collects sensitive data and hide them on the HD.
- Just plug a USB device with autorun-like capability and collect the data hidden.
- Other possibilities with steganography:
  - The malware hides secrets to evade in normal data only when legitimately copied to a USB device.
  - The USB device can be that of the legitimate user. Just steal or dump the key.
- In both cases, a contact with the target computer is required.
  - Very efficient techniques!



# About steganography

- Recall: hiding secrets by embedding them into a innocent-looking document (JPEG, PDF, EXE files, MP3…).
  - Secret data + cover-medium = stego-medium.
- Supposed to be used by Al-Qaida in 2001 and later.
- To extract the information, you need to have access to the stego medium.
- In our context, this is not applicable:
  - No network or other direct access.
  - The stego-medium can be modified by noise.
- So you have to consider a different kind of steganography.


esiea

# Our first operational technique

## The Windows Jingle Attack

# Principle of the attack

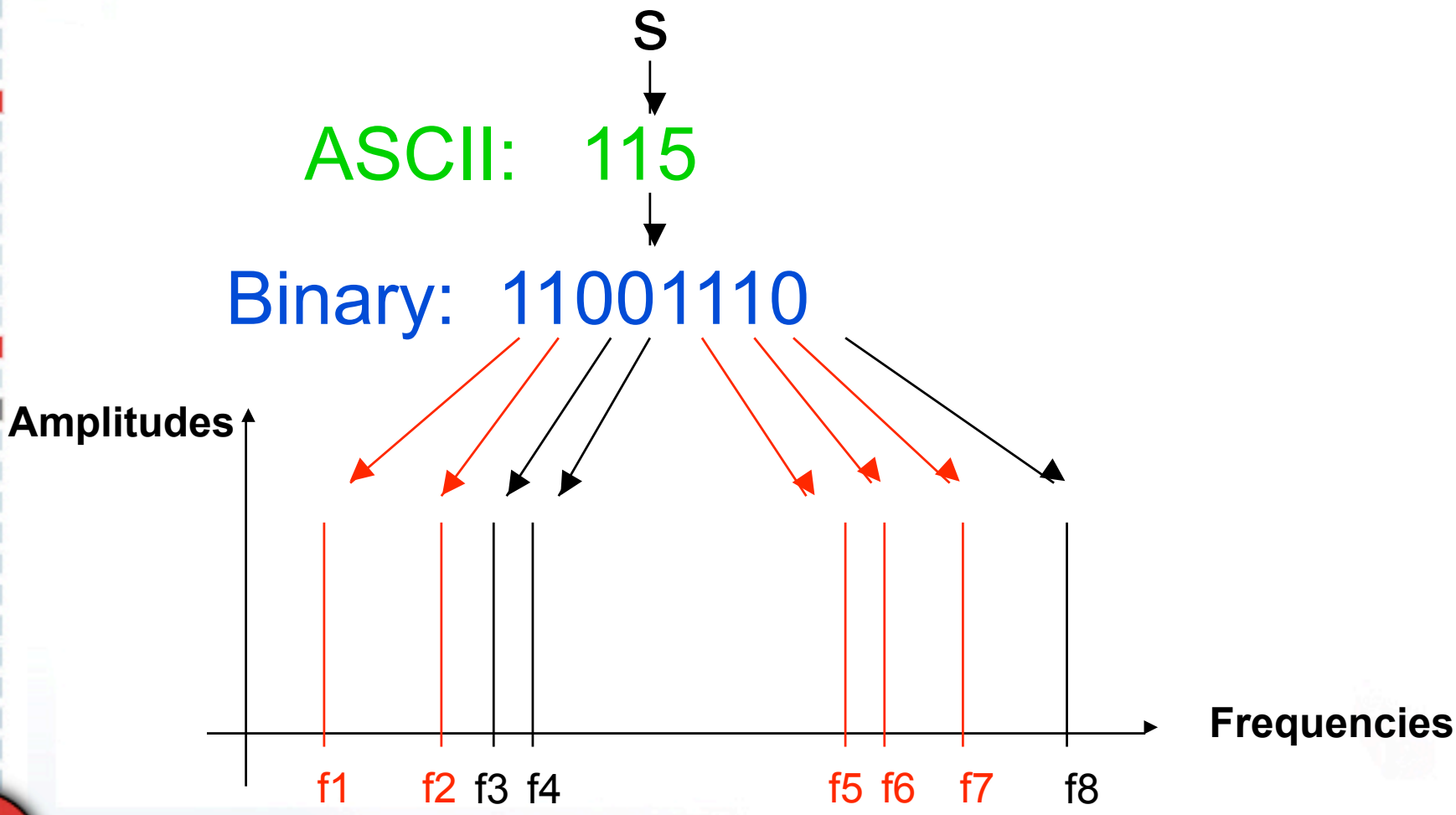
- The malware collects the secret (e.g. login/passwd).
- Use sort of audio steganography to hide it into the Windows (opening) session jingle (lots of other possibilities, e.g. user's MP3 files).
- Intercept the Jingle:
  - Local hidden microphone.
  - Long distance remote sensitive microphone or laser-based microphone.
- Extract and decode the data.
- Passive technique. Can be generalized to files.


esiea

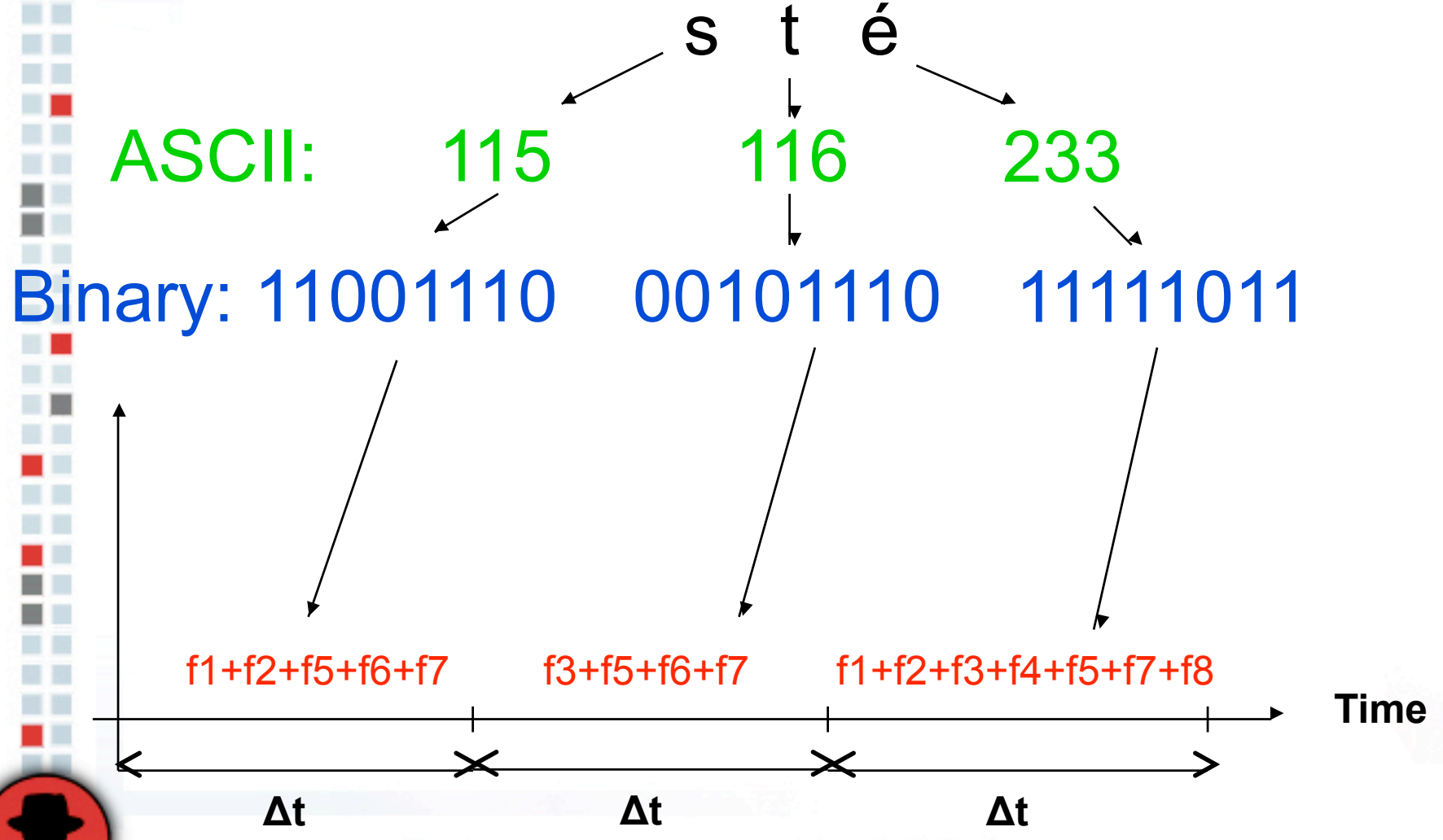
# The tools

- Tools used to validate the attack:
  - Scilab for the signal processing with Fourier tranforms (coding/decoding).
  - An audio software (recording the audio signal).
  - A microphone.
- The Scilab part can be easily implemented into a malware (basic signal processing algorithms).

# Encoding the secret



# Encoding the secret #2



# Constraints

To transmit 20 characters (login/passwd).

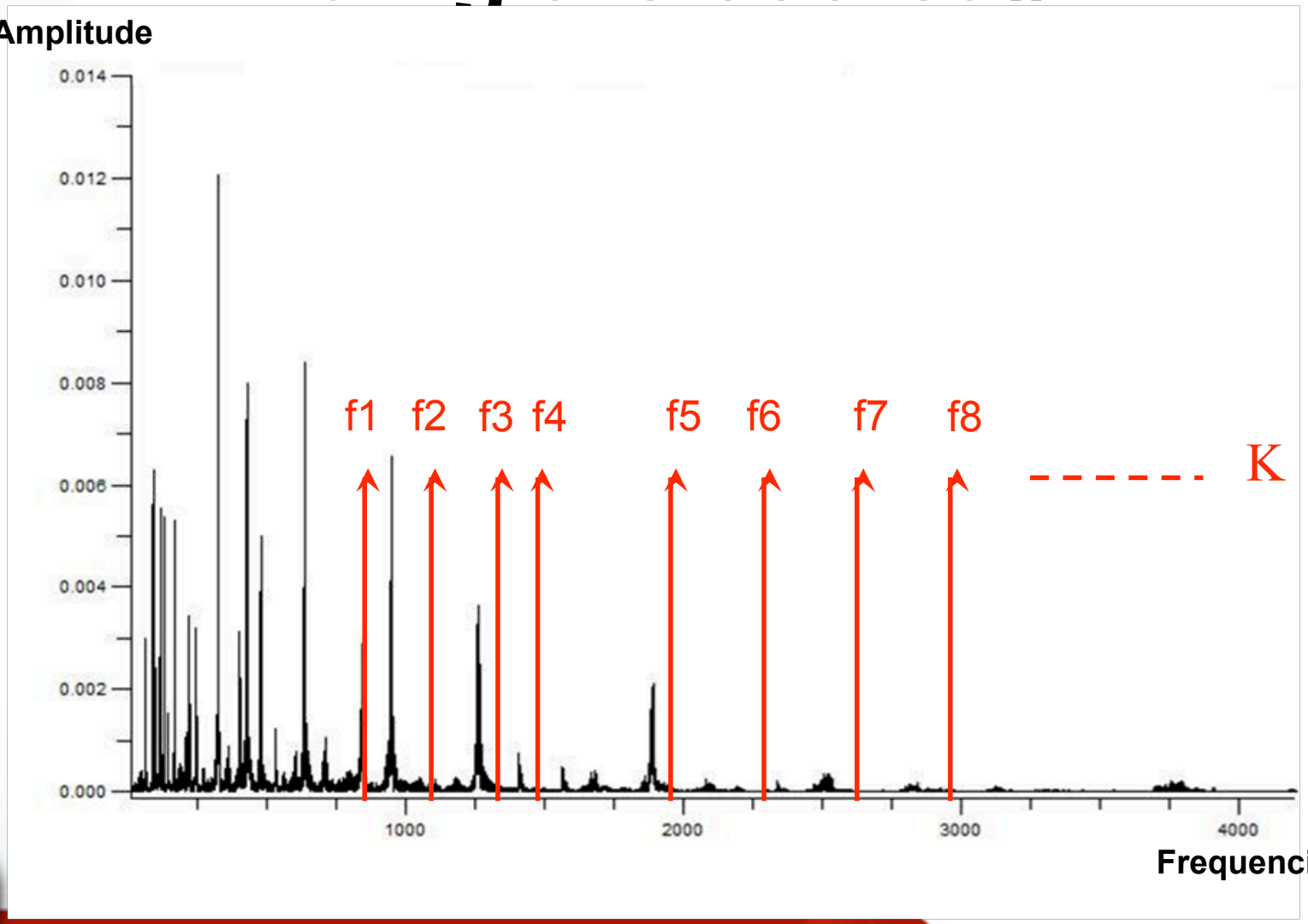
- Be as stealth as possible:
  - Use an amplification coefficient K as limited as possible.
- Adapt to the Windows Jingle length (5 sec.)
  - 20 * $\Delta t$ < 5 secondes.
- Audio tone S(t) of freq. $f_i$
  - $S(t) = K * \cos(2\pi * f_i * t)$
- Let us first validate the approach by considering the secret « ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ » (worst case).

# Hiding the secret

Secret : « ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ »

| | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | | ÿ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Freq. 1 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 2 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 3 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 4 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 5 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 6 → | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. |
| Freq. 7 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |
| Freq. 8 → | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 0. |

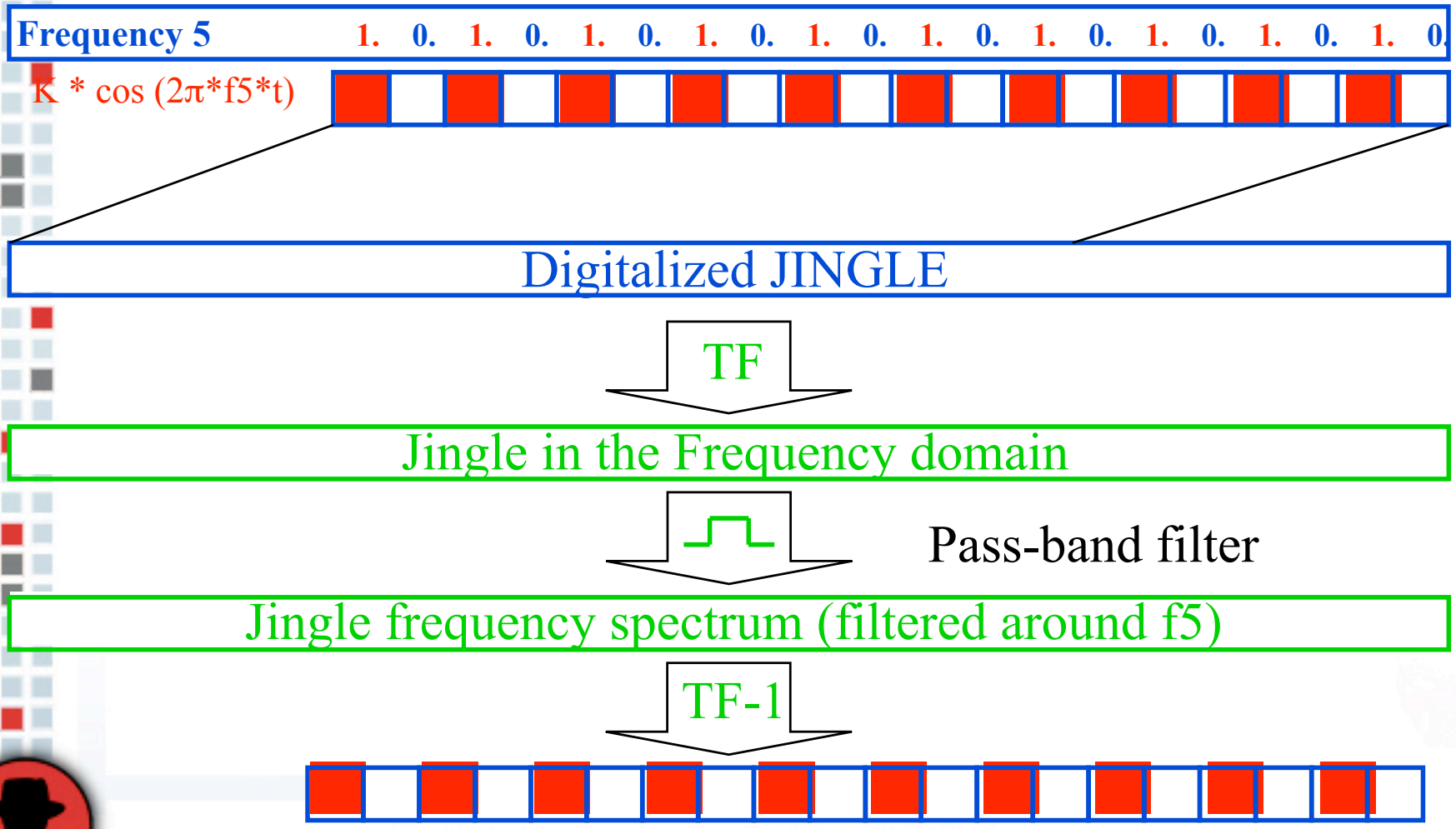
# Hiding the secret #2



# Hiding the secret: choosing the encoding frequencies

- Encoding frequencies must be chosen carefully.
- The lower limit is imposed by the computer speakers.
  - Must transmit the sound with a good quality. Most speakers are unable to play sounds at too low frequencies.
- The upper limit is defined by the Shannon sampling theorem
  - For our sampling frequency (F = 22050 Hz), this gives a maximal frequency of F/2 = 11 KHz.
- Possible frequency choice:
  - Finding the best trade-off between values close to Windows Jingle frequency peaks and values corresponding to flat frequency aeras.
- Best frequencies with respect to the Windows Jingle:
  - 900, 1100, 1350, 1500, 2000, 2250, 2700 and 3000 Hz.
- Other choices are possible and be randomly chosen according to a scheme known by the attacker only.

# Hiding the secret #3



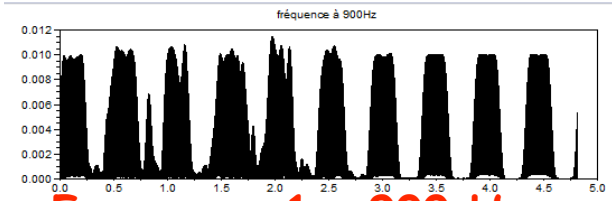


# Hiding the secret: steps

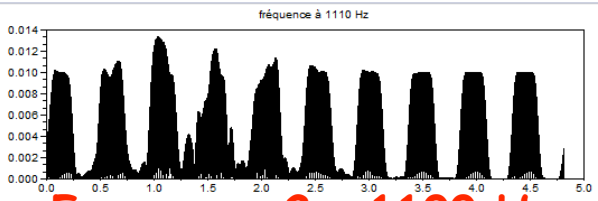
- Encode the binary secret as frequencies.
- Emission of the frequency in the time domain:
  - Alernative emission during $\Delta t$ followed by a non emission during $\Delta t$…
  - Add the encoded secret to the Windows Jingle.
- To extract the secret from the Jingle:
  - Apply a Fourier transform (frequency domain).
  - Filter around the suitable carrier frequency (BPF).
  - Apply an inverse Fourier transform (time domain).
  - Look for the corresponding carrier frequency to be present (bit 1) or not (bit 0).

# Extracting the secret

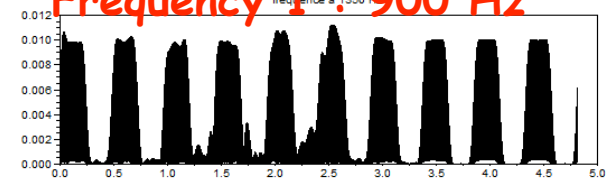
- Secret: ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ



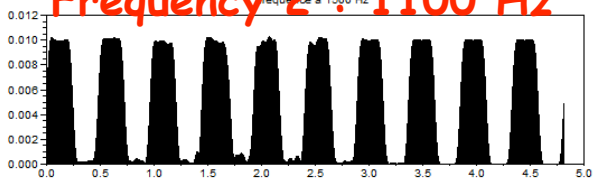

Frequency 1 : 900 Hz



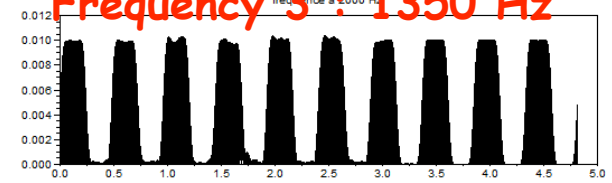

Frequency 2 : 1100 Hz



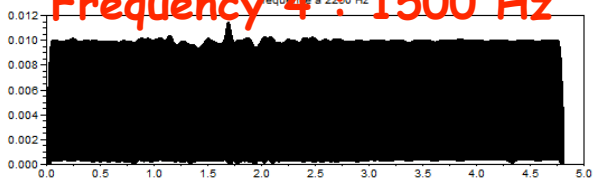
Frequency 3 : 1350 Hz



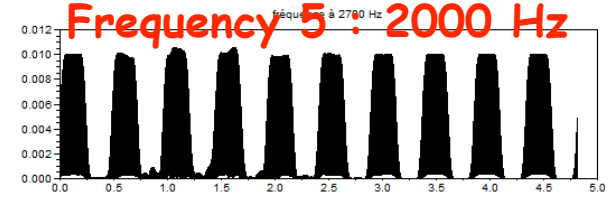
Frequency 4 : 1500 Hz



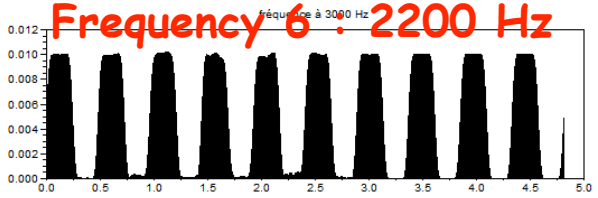
Frequency 5 : 2000 Hz



Frequency 6 : 2200 Hz



Frequency 7 : 2700 Hz



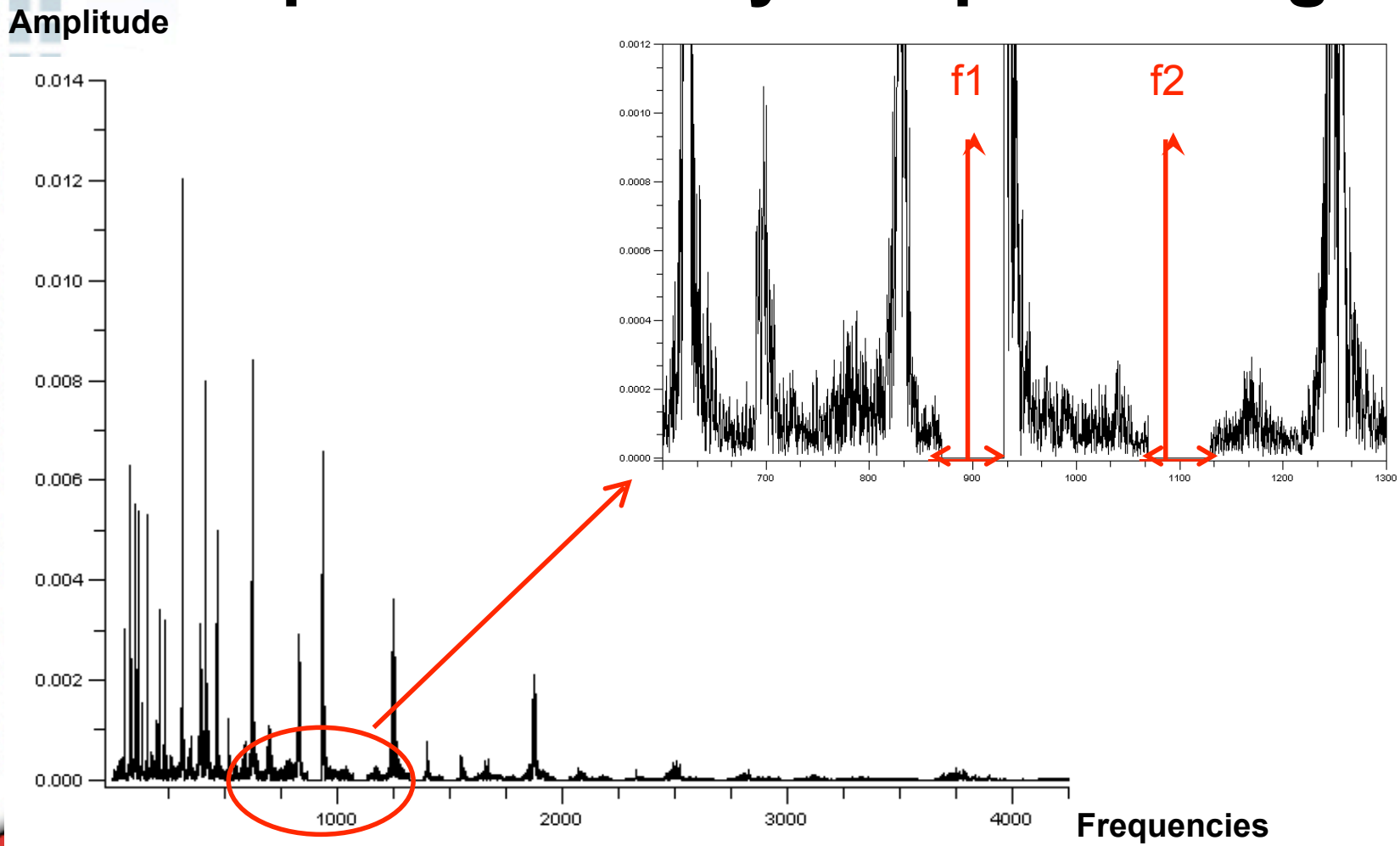
Frequency 8 : 3000 Hz


esiea

# Optimisation by Freq. Cleaning

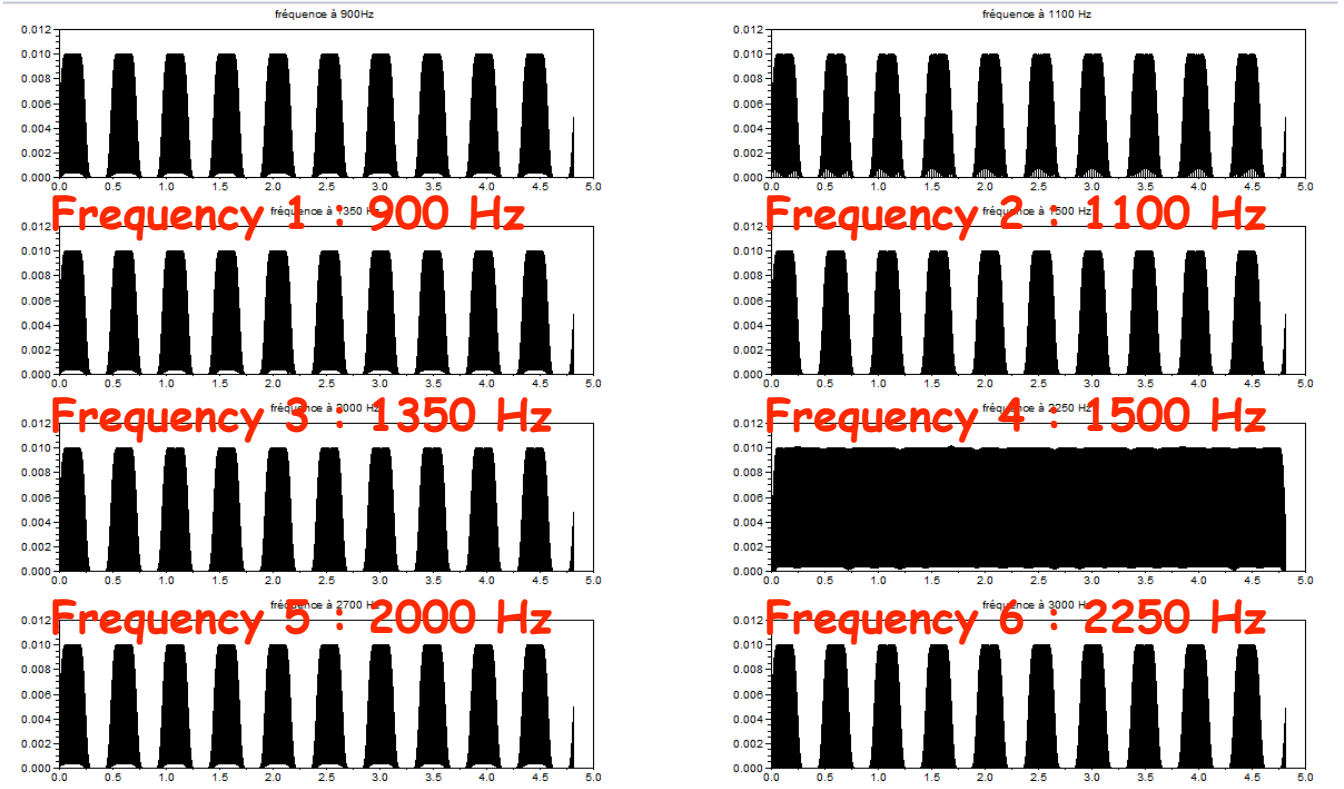
- To bypass decoding residual errors (due to frequency interferences), we have designed the frequency cleaning technique:
  - Just erase (filter) Jingle parasitic frequencies around our encoding frequencies to cancel those interferences.
- Produces a far better decoding with no residual errors while preserving a stealth transmission of the secret.

# Optimisation by Freq. Cleaning #2



# Optimisation by Freq. Cleaning #2

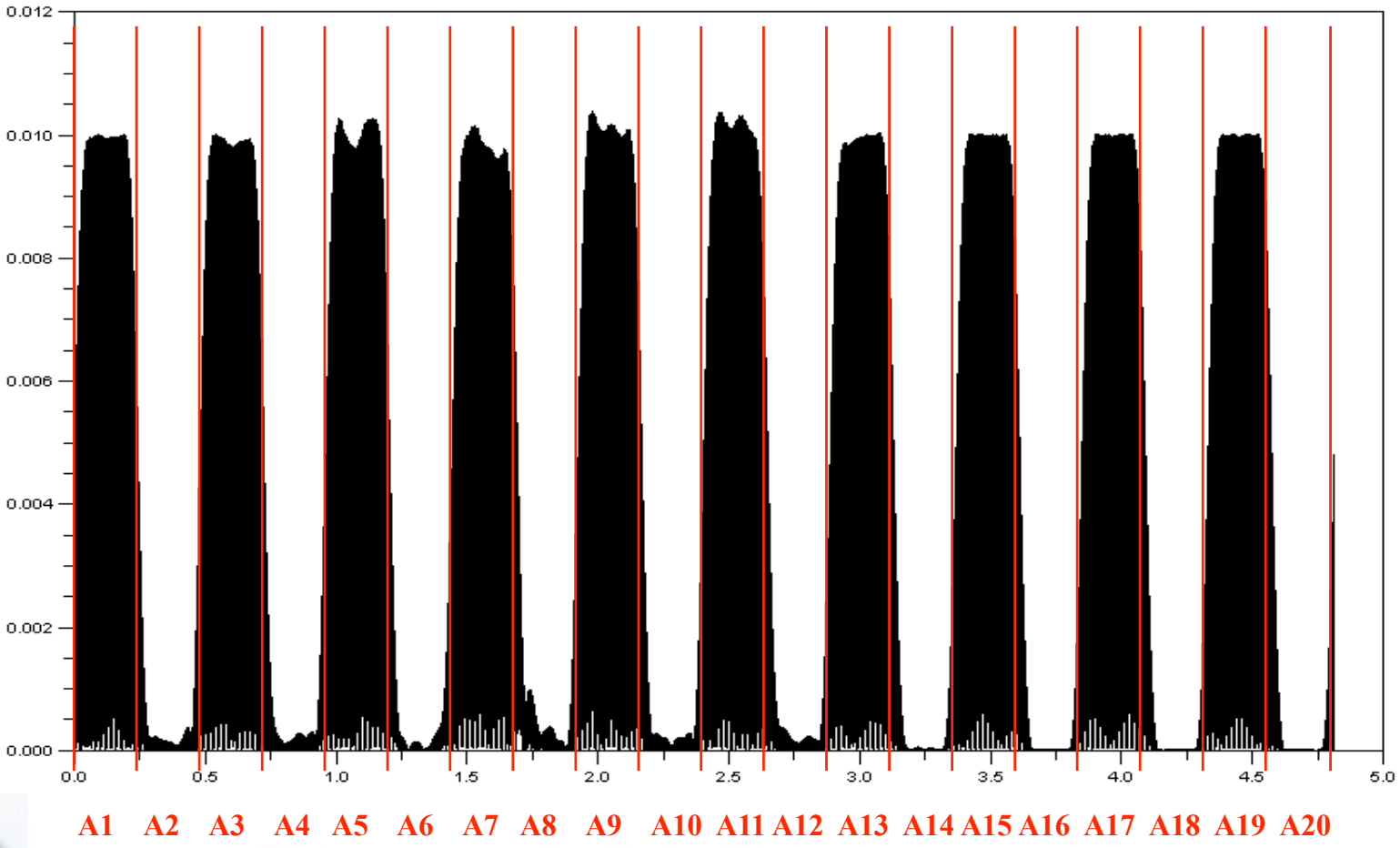
- Secret : ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ ÿ



# Extracting the secret #2

- To decide between 0 and 1, we consider the time representation of the encoding signal:
  - Compute the different surfaces $S_1$, $S_2$…under the curve with respect to time intervals of length $\Delta$ (areas $A_1$, $A_2$…).
  - Compute the ratios $R = S_i/\hat{S}$ where $\hat{S}$ is the maximal possible surface under the curve.
  - If R is « close » to 1 decide bit = 1 otherwise if R is « close » to 0 decide bit = 0
  - A decision threshold is required (see further).

# Extracting the secret #3





# Experimental validation

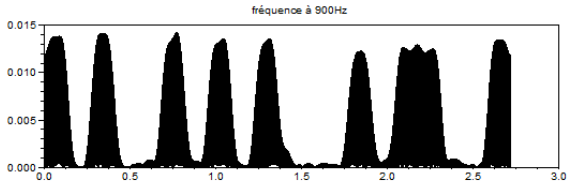
Secret: « stéphane et vincent! »

| | s | t | é | p | h | a | n | e | | e | t | | v | i | n | c | e | n | t | ! |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Freq. 1 → | 1. | 0. | 1. | 0. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 0. | 0. | 1. | 0. | 1. | 1. | 0. | 0. | 1. |
| Freq. 2 → | 1. | 0. | 0. | 0. | 0. | 0. | 1. | 0. | 0. | 0. | 0. | 0. | 1. | 0. | 1. | 1. | 0. | 1. | 0. | 0. |
| Freq. 3 → | 0. | 1. | 0. | 0. | 0. | 0. | 1. | 1. | 0. | 1. | 1. | 0. | 1. | 0. | 1. | 0. | 1. | 1. | 1. | 0. |
| Freq. 4 → | 0. | 0. | 1. | 0. | 1. | 0. | 1. | 0. | 0. | 0. | 0. | 0. | 0. | 1. | 1. | 0. | 0. | 1. | 0. | 0. |
| Freq. 5 → | 1. | 1. | 0. | 1. | 0. | 0. | 0. | 0. | 0. | 0. | 1. | 0. | 1. | 0. | 0. | 0. | 0. | 0. | 1. | 0. |
| Freq. 6 → | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. |
| Freq. 7 → | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 0. | 1. | 1. | 0. | 1. | 1. | 1. | 1. | 1. | 1. | 1. | 0. |
| Freq. 8 → | 0. | 0. | 1. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0. |

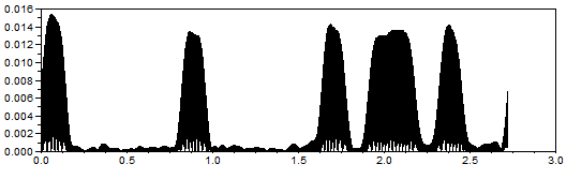
# Experimental validation #2

Secret : « stéphane et vincent! » (K=0.01)

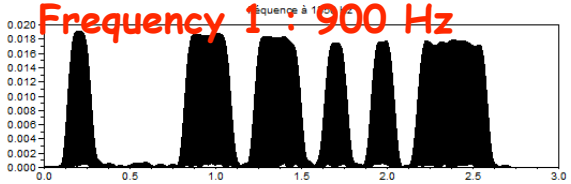
Secret extracted : « stéphane et vincent! »



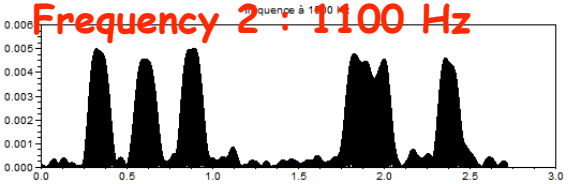

Frequency 1 : 900 Hz



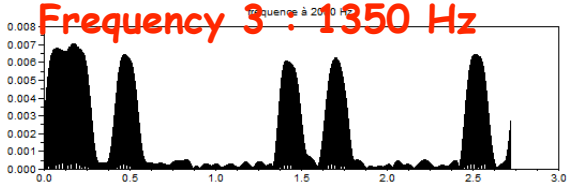

Frequency 2 : 1100 Hz



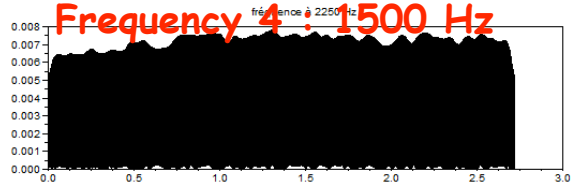
Frequency 3 : 1350 Hz



Frequency 4 : 1500 Hz



Frequency 5 : 2000 Hz



Frequency 6 : 2250 Hz

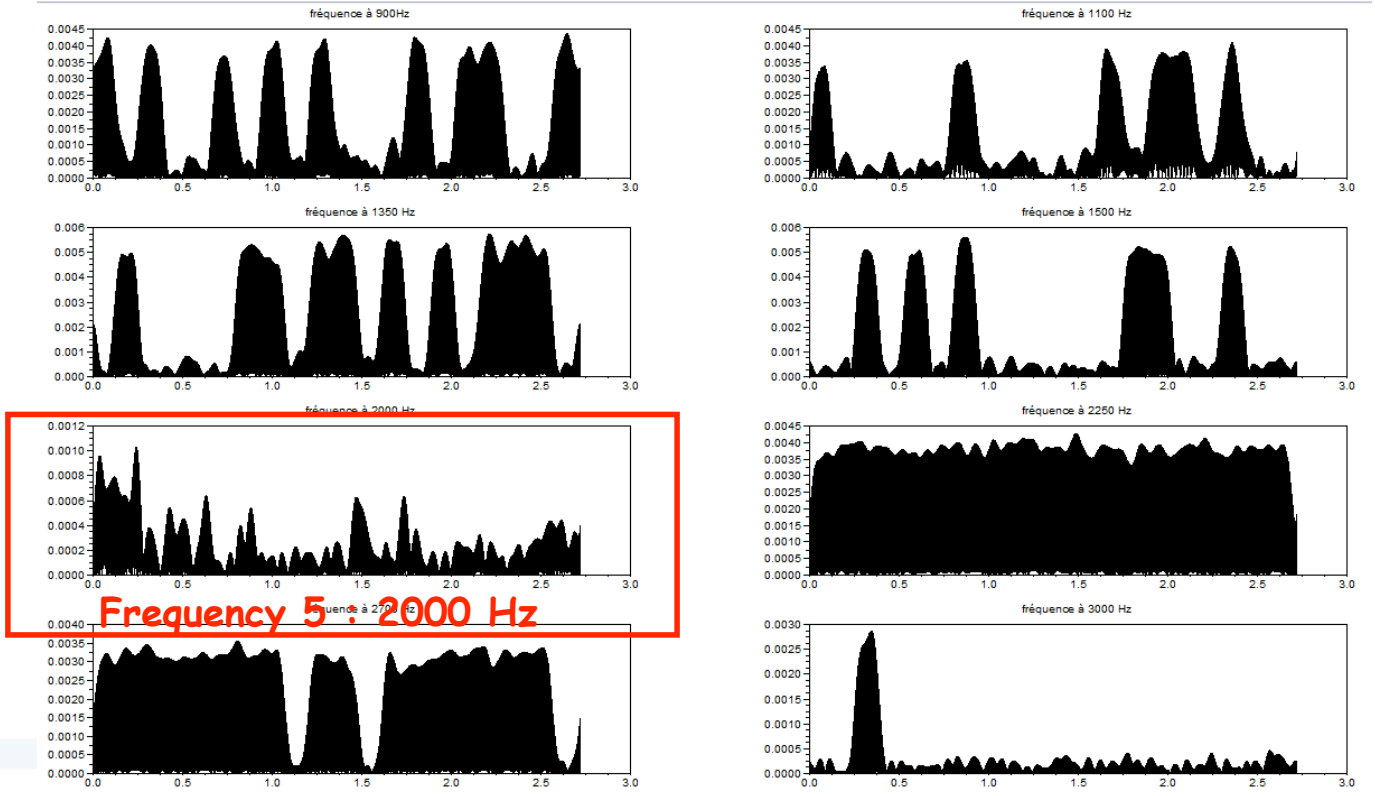
Frequency 7 : 2700 Hz

Frequency 8 : 3000 Hz

# Experimental validation #3

Secret: « stéphane et vincent! » (K=0.003)

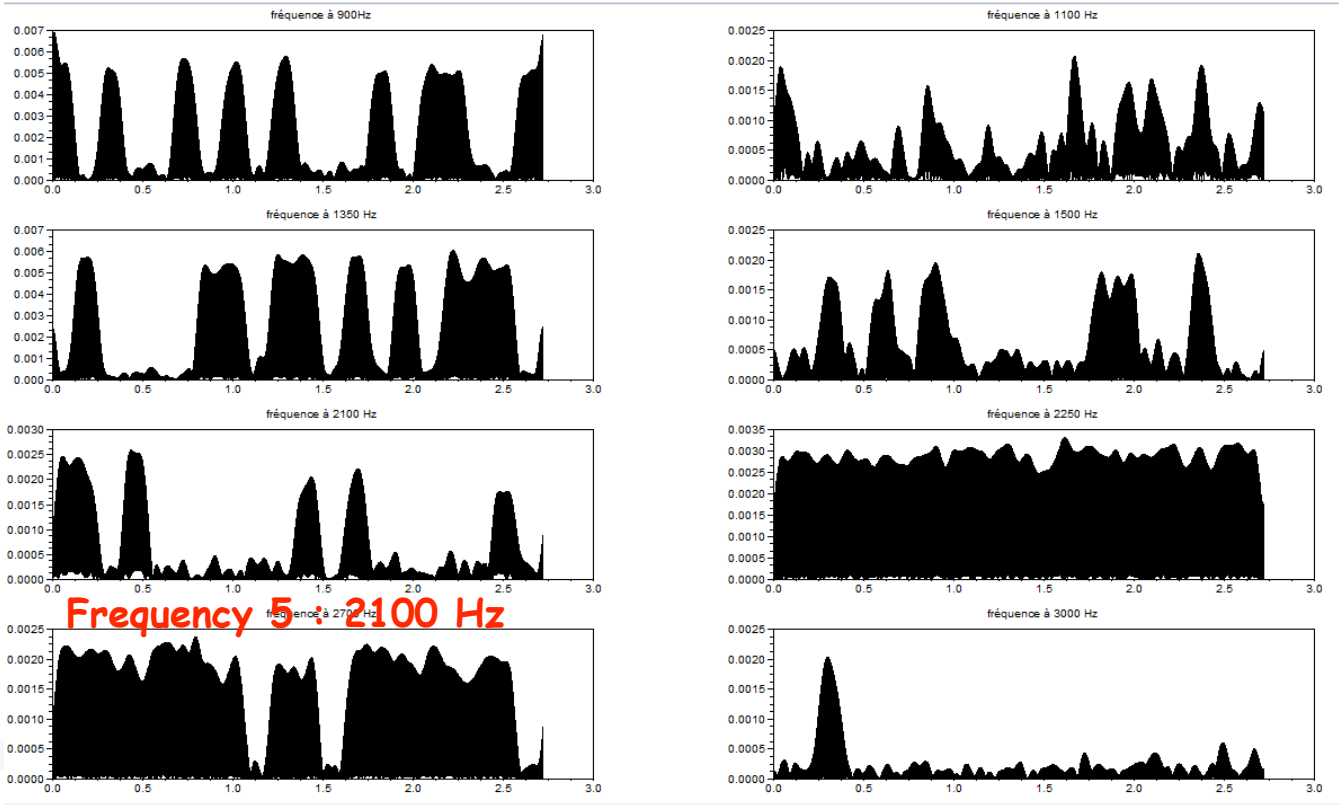
Secret extracted: « cdé`hane ed fincend! »



# Experimental validation #4

Secret: « stéphane et vincent! » (K=0.003)

Secret extracted: « stéphane et vincent! »

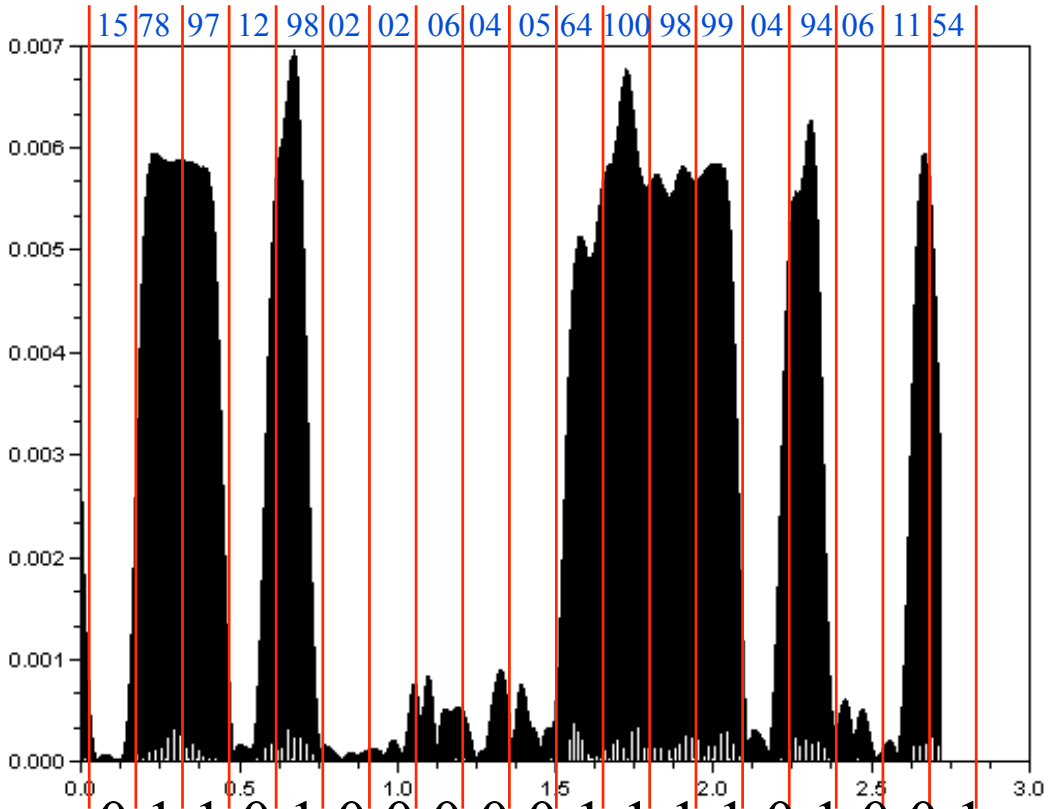


# Experimental validation #5

- Short demos:
  - The original Windows Jingle
  - The Jingle with hidden data (K = 0.01).
  - The Jingle with hidden data (K = 0.003).
  - The Jingle with hidden data (K = 0.001).
- Challenge:
  - Will be made available on the BH website (archives).
  - A very (too) simple example.
  - Extract the secret data, send the solution and you win!

# Optimizations

Surface/Surface max (%)

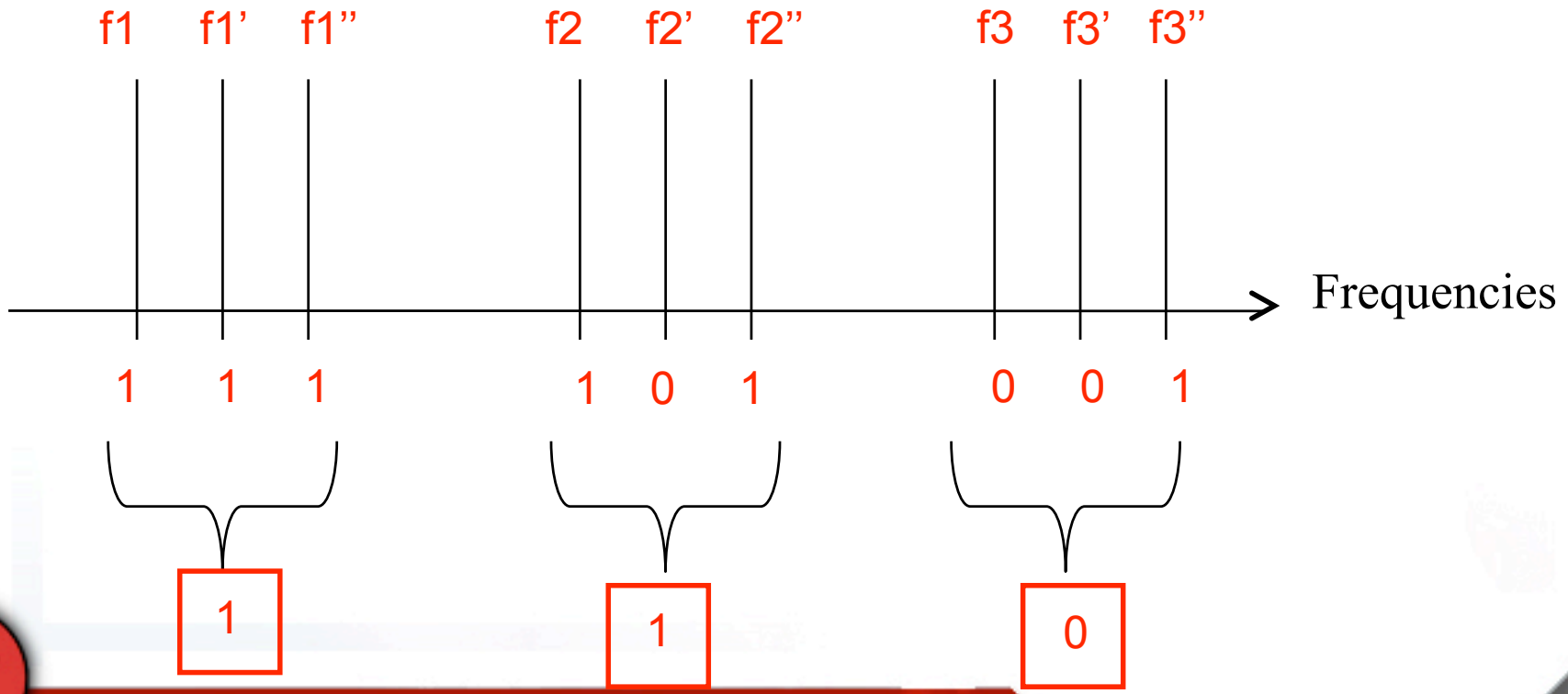



**Threshold = 50 %** 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 1

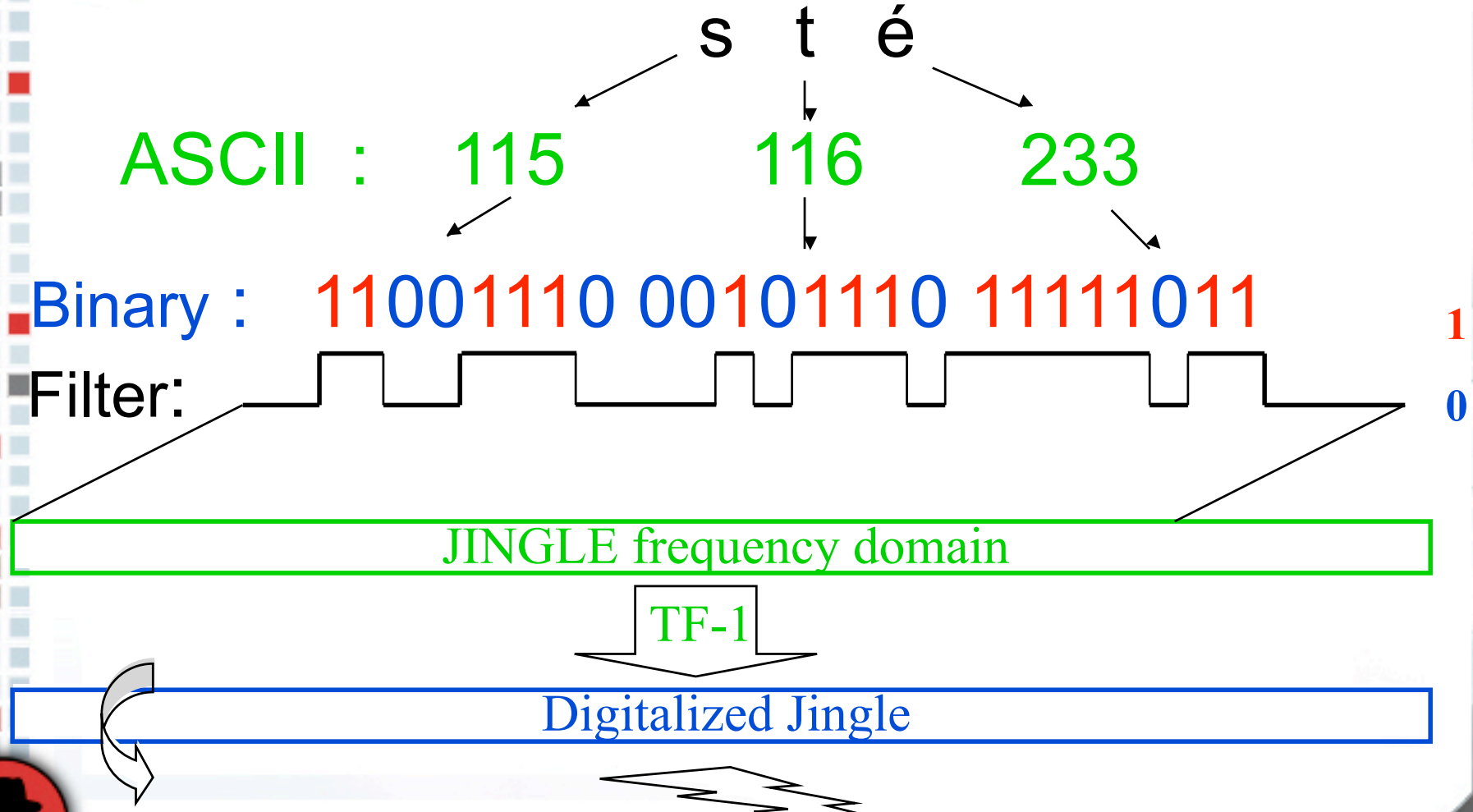
**T = 33 % - 66 %** 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 1

# Optimizations #2

- Use error-correcting techniques.
- Example: (n, 1, $\frac{n-1}{2}$) repetition codes.



# Optimizations #3: cavity encoding



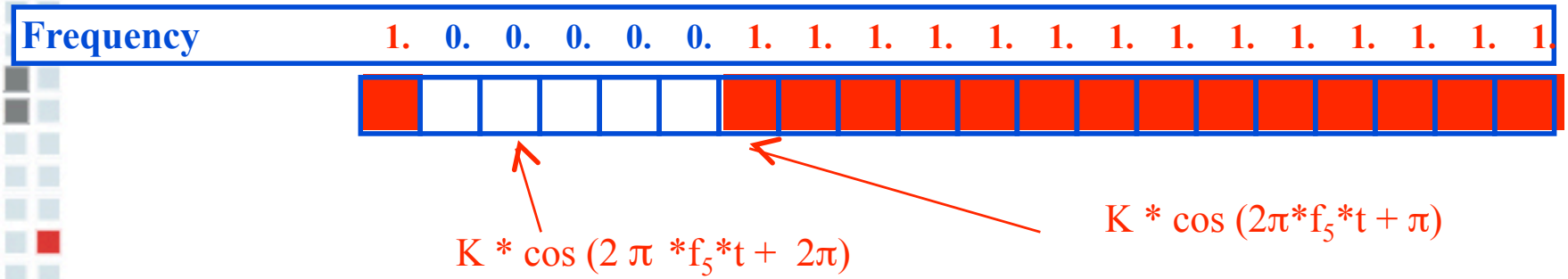

esiea

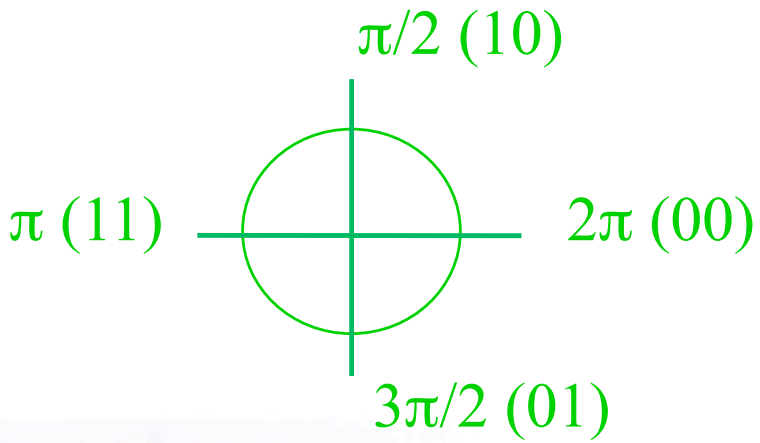
# Optimization #4: phase shift keying

- Use native frequencies in the jingle.
- First method: consider two phase displacements $\pi/2$ and $\pi$ (to encode respectively 0 and 1).
- Second method: use more phase displacements to increase the transmission rate.
  - With QPSK we encode two bits at a time.
- More efficient, yet tricky, methods.

# Optimizations #4

## PSK method



# To conclude with the Jingle attacks

- Very efficient attack.
- Reasonably good interception at several ten meters.
- Many other optimizations are possible and are under current development.
- Only limited tools required:
  - A simple malware can performed it!
  - Accessible to many attackers.
- Totally stealth technique.
- Other sound files can be considered.
- Other frequencies ranges (e.g. 20 – 25 KHz).


esiea

# Our other attacks

**or**

**« Operational solution are often the most simple one ».**

{Visual, HD, Fan} covert channels



# General Principle

- Many computer hardware devices behave differently according the OS/processor activity.
  - HD noise on read/write access.
  - FAN noise on processor load.
  - Monitor on/off.
  - Windows reboot.
  - ….
- Use those variable behaviours to encode secret data.
- Interception can be performed at a rather long distance.

# General Principle #2

- Code the secret to make evade in hexadecimal.
- Define a one-to-one mapping between the 16 possible hexa characters and time interval values.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $t_0$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ |
| 8 | 9 | A | B | C | D | E | F |
| $t_8$ | $t_9$ | $t_A$ | $t_B$ | $t_C$ | $t_D$ | $t_E$ | $t_F$ |

- The time interval value will depend on the event or harware resiliency .

# Visual covert channel

- As a nice example: monitor activity.
- Use Windows API primitives to manage monitor power.
  - ✓ The malware switches between power off and power on to code the secret.


Time interval $t_i$ to code nibble i

# Visual covert channel #2

- Works fine night and day at a potential long distance:
  - Day: use field glasses.
  - Night: use night vision goggles.
- Many other possibilities:
  - Screensaver, shutdown, reboot…
- Do not forget to adapt the time value $t_i$ to the used device/event natural latency for an optimal decoding.
- The malware must « transmit » during suitable periods:
  - lunch time, night, user absence…

# Audio covert channel

- We can apply the same approach in the audio domain:
  - Changing the FAN speed results in a different FAN noise.
  - the malware just manipulate the FAN speed at different time intervals to code the secret.
  - *SetSpeed* Method (*CIM_FAN Class*).
  - Further developments to come...
- Read/write activity for most of the case will generate HD noise (e.g. clicks).
  - the malware just read/write random data on the HD at different time intervals to code the secret.
  - A little demo…

# Audio covert channel #2

- More complex to implement but really efficient.
- With audio covert channels, we must consider sophisticated error-correcting methods to encode the secret data.
  - This is necessary to get rid of parasitic ambient noises during the data extraction.
- Combining different devices should be far more powerful.
- The sequence of devices used is known by the attacker only (equivalent to a secret key).



# Conclusion

- Making data evade from a non networked computer is easier than expected:
  - Just let your imagination play.
  - Potentially many other possibilities.
    - E.g. {webcam, network cards…} leds.
    - Electric current consumption.
  - Malware can do the encoding very easily.
- What about networked computers?
  - Bad news: it can be worse!
  - IPSec subvertion (Filiol, Jennequin, Delaunay – ECIW 2008).


esiea

# Conclusion #2

- The only limitation is not the interception part:
  - There exist many highly sensitive microphones.
  - Interception can be performed far away from the target computer.
- The tricky part is the signal processing to extract the secret.
  - Not a problem for intelligence (private or not) agencies or for a good, well-equiped sound engineer!



# Conclusion #3

- So what is the solution?
  - Only one…
  - … avoid to be infected by malware!
  - Efficient security policy + antivirus.
- Or don't use computer at all!


esiea

# Bibliography


- W. van Eck (1985), Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computers and Security 4*.
- H. Tanaka, O. Takizawa and A.Yamamura (2004), Evaluation and Improvement of Tempest Fonts, *Information Security Applications (WISA 2004)*, LNCS #3325, pp. 457 – 469, Springer-Verlag.
- H. Tanaka (2008) Information Leakage via Electromagnetic Emanations and Evaluation of Tempest Countermeasures. *National Institute of Information and Communications Technologies*, Japan, to appear.
- http://camouflage.unfiction.com/
- http://assiste.com.free.fr/p/abc/a/podslurping.html/
- http://www.usbhacks.com/2006/10/29/how-to-simple-podslurping-example-with-a-usb-flash-drive/




# Bibliography #2


- M. Kuhn (2003), Compromising Emanations: Eavesdropping Risks of Computer Displays, *Technical report UCAM-CL-TR-577*, U. of Cambridge.
- M. G. Kuhn and R. J. Anderson (1998), Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations, *Information Hiding 1998 (IH'98)*, LNCS #1525, pp. 124 – 142, Springer-Verlag.
- M. Kuhn, Filtered-tempest Fonts, available at http://www.c1.cam.ac.uk/ mgk25/stfonts.zip/
- M. Kuhn (2004), Electromagnetic Eavesdropping Risks of Flat-Panel Displays, *Privacy Enhancing Technologies 2004*, LNCS #3424, pp. 88 – 107, Springer-Verlag.
- http://www.eskimo.com/~joelm/tempest.html
- E. Filiol, F. Jennequin and G. Delaunay (2008) Malware-based Information Leakage over IPSec Tunnels. *7th European Conference on Information Warfare and Security*, Plymouth, June 2008.



esiea

Thanks for your attention

Questions ?