

# Bypassing Data Exfiltration Detection With Malicious Cryptography Techniques

Eric Filiol

[efiliol@hse.ru](mailto:efiliol@hse.ru), [contact\\_ops4sec@pm.me](mailto:contact_ops4sec@pm.me)

<https://ericfiliol.site>

OPS4SEC, Estonia & HSE, Moscow, Russia



NATIONAL RESEARCH  
UNIVERSITY



# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion

# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion

# Introduction: Key Security Issue

- Most attacks now include data exfiltration from the target.
  - From a few credentials...
  - ... to databases.
- Depending on the environment, it is more or less difficult for the attacker
  - Unconnected environments require air-gap attacks  $\Rightarrow$  limited amount of data, low data rate, low security awareness.
  - Connected/networked environments (any protocol) require bypassing traffic surveillance  $\Rightarrow$  Possibly high amount of data, high data rate and medium/high security awareness.
- From the defender perspective, the analysis and safeguards are
  - Automated or semi-automated analysis
  - Manual/ad hoc analysis

# Introduction: Working Environment & Scenarios

- Our operational context is twofold:
  - Either relatively weakly targeted attacks for environments where only automated detection is likely to be in place.
  - Or strongly targeted attacks (single or very few target) where a manual analysis of malware/traffic data can be performed.
- We focus on connected environments (network with any protocol) but our techniques apply to air-gap environments as well.
- If we primarily focus on bypassing automated analysis, we cannot neglect the ad hoc/manual analysis.
  - The initial step consists in deploying a malware (most of the times) or a dedicated device
  - We have consequently to manage this analysis step as well. This analysis must not reveal the actual nature of the attack (and ultimately that an attack is under way) at much as possible.

# Introduction: Working Environment & Scenarios (continued)

- We focus on and deal with communication channels in place only!
- Creating alternative/new communication channels is another issue and is out of scope of the present talk
  - Airgap attacks, covert channels, invisible channels...
  - Drop me an email if you wish references.

# Attacker's Issues to Solve

- The attacker has to face several critical issues than can trigger alerts and block data exfiltration :
  - 11 Data may be analysed so semantic detection (keywords, statistical profile, Data Leak Prevention [DLP]) can be enforced.
  - 12 Encrypting data before exfiltration is likely to be detected by a simple entropy profile test (yet it is rarely in place)
  - 13 Encryption means a secret key that can be recovered during malware analysis or during the process performing the data exfiltration.
  - 14 All outbound traffics may be encrypted automatically (IPSec VPN) and this encryption it out-of-control for this attacker (this is the case in military networks for instance)
- Any analysis by the defender must fail or at least be delayed enough.

# Aim of the Present Work

- Showing how an attacker could exfiltrate sensitive data while bypassing all these issues by using malicious cryptography & mathematics.
- Evaluating forthcoming approach by malware designers/attackers to make malware/ransomware techniques evolve in a more critical way.
- Identify and test possible mitigation techniques to prevent the presented techniques in a proactive way (precautionary principle)
- We present “unitary attack bricks” for clarity but they can be combined in whole or in part.
- A detailed **TLP:WHITE** paper will published soon (on [arxiv.org](https://arxiv.org) repository).
  - All codes and PoC developed are **TLP:RED**



# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion

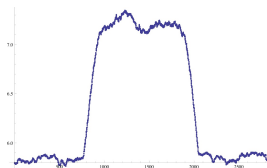
# ComSec versus TranSec

Two different views in Information Security (NATO terminology)

- **ComSec** (Communications Security) ensures the security (confidentiality and integrity) of telecommunications. In other words, ComSec refer to the security of information that is transmitted or communicated regardless of the communication channel.
  - Cryptography, Tempest/EMSEC, physical security (network, rooms...)
- **TranSec** (Transmission Security) ensures the protection of the channel itself and especially the existence of secret data being exchanged (prevent interception, disruption of reception, communications deception, and/or derivation of intelligence by analysis of transmission characteristics such as signal parameters or message externals...).
  - TranSec is a field of COMSEC which deals with the security of communication transmissions (the channel), rather than that of the information being communicated.
  - Steganography, Tempest/EMSEC, traffic flow security, routing protocols...
- Our techniques intends to combine both views.

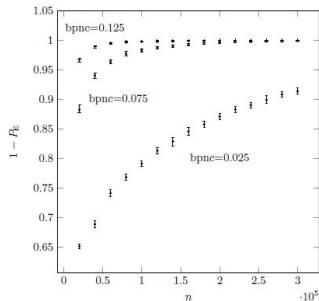


# Cryptography versus Steganography: detection



- Detection of cryptography is straightforward using the entropy profile [2].

- Plaintext data: entropy  $H(X) \approx 4$
- Packed/compressed data:  $H(X) \approx 6$
- Encryption data:  $H(X) = 8$



- Beyond an insertion rate of 0.03, detection is efficient with modern techniques [3]
- Size of secure payload is limited (to  $\sqrt{n}$ ;  $n$  is the number of usable coefficients for embedding).

# Malicious Cryptology & Mathematics

- Malicious Cryptology and Malicious Mathematics (MCMM) is an emerging domain initiated in (Filiol, 2012)
  - Generalization of Young & Yung's (2004) crypto virology Young (limited case of extortion malware which prefigures ransomware).
- MCMM can be defined as the interconnection of attack techniques with cryptology and mathematics for their mutual benefit. Covers several fields and topics (non exhaustive list):
  - Development “super malware” able to evade any kind of detection by implementing:
    - Optimized propagation and attack techniques (e.g. by using biased or specific random number generator).
    - Sophisticated self-protection techniques. The malware code protects itself and its own functional activity by using strong cryptography-based tools.
    - Partial or total invisibility features. The programmer intends to make his code to become invisible by using statistical simulability

# Malicious Cryptology & Mathematics (continued)

- Use of complexity theory or computability theory to design undetectable malware.
- Use of malware to perform cryptanalysis operations (steal secret keys or passwords), manipulate encryption algorithms to weaken them on the fly in the target computer memory. The resulting encryption process will be easier to break/bypass.
- Recon in target environments (e.g. processor-dependent malware)
- Design and implementation of encryption systems with hidden mathematical trapdoors. The knowledge of the trap (by the system designer only) enables to break the system very efficiently. Despite the fact that the system is open and public, the trapdoor must remain undetectable (see a real instance in [Filiol & Banner, BlackHat Europe, 2017]).

See bibliography slide for extended references [1].

# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography**
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion

# Non-Trivial Deniable Cryptography: Principles

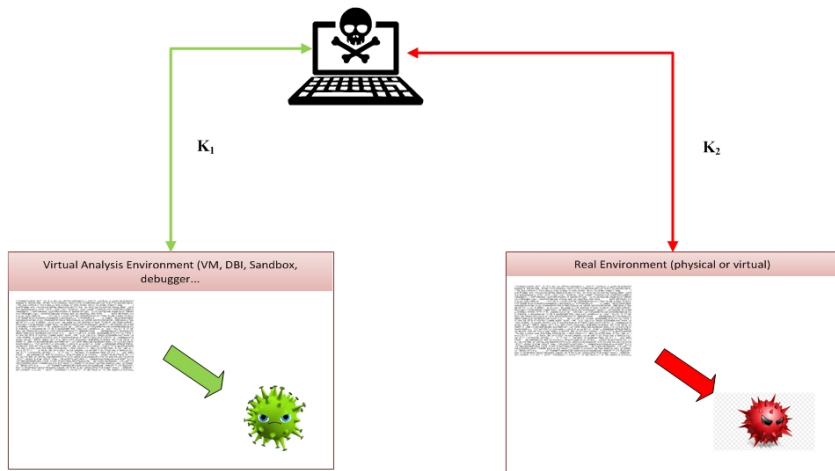
- Building effective encryption algorithm (deterministic algorithms) to realize practical deniable cryptography was until very recently still an open problem.
  - Only known case is trivial (one-time pads).
  - Since the “key” is as long as the two (or more) plaintexts, this solution is not valid (one-time must be in the code).
- Let  $C$  be a ciphertext of length  $N$ , a unique algorithm  $E$  and any two different arbitrary plaintexts  $P_1$  and  $P_2$ . We built a C framework to build encryption algorithms (within seconds from given plaintexts) enabling effective deniable cryptography with short keys (128 - 256 bits).
  - $E$  is a deterministic encryption algorithm (stream cipher or block cipher). It is supposed to be public and therefore resistant to known cryptanalysis techniques. Keys are  $k$ -bit long.
  - $k$  is far smaller than  $N$  (so one-time pad is not considered).
  - We have  $C = E(K_1, P_1) = E(K_2, P_2)$
  - The scheme can be extended to a finite number of plaintexts  $P_i$



# Non-Trivial Deniable Cryptography: Applications

- The cryptographic security analysis of these algorithms have confirmed the resistance against the following attacks:
  - Guess  $P_1$  and  $P_2$  from the ciphertext  $C$  (in other words, retrieving keys  $K_1$  and/or  $K_2$ ).
  - Find  $P_1$  knowing  $P_2$  and conversely.
- **Awesome number of applications:**
  - Code protection (malware or legitimate program) against static and dynamic analysis (see next slide).
  - Anti-forensics techniques.
  - Multiple communication channels in a single one (flow deniable cryptography).
  - ...
- **Demo**

# Deniable Cryptography-based Malware



# Deniable Cryptography-based Malware

- The most critical part of the malware is encrypted ( $C$ ) and needs an external key from the C&C .
- Secure and complex communication protocol between malware and C&C (fingerprint, time index, time obfuscation, random connexions, environment conditions...). The malware is clueless wrt this protocol.
- The malware is able to detect that it is under analysis (see [4] for instance)
- The malware analyses its environment and requests an external key according to the connexion protocol.
- If no analysis is under way, the malware receives key  $K_2$  and then decrypts itself as  $P_2 = E(K_2, C)$ . This the real malware.
- If analysis is detected and/or connexion conditions are not fulfilled, the malware receives key  $K_1$  and then decrypts itself as  $P_1 = E(K_1, C)$ . This is either a goodware or an alternative malware (to fool the malware analyst).

# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking**
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion

# Metadata & Document Formats/Internals Permissiveness

- Most documents formats includes metadata and rather rich/complex formats/internals:
  - Most DLPs do not check metadata or are very weak (easy to bypass) at analysing them.
  - Underlying format languages/internal either are not properly specified or the compliance to the internals is partially or not checked.
- Most document formats have a large permissiveness with respect to data embedding.
- Depending on the document format, it is possible to silently exfiltrate from a few tens of bytes to several megabytes
  - Documents are extremely interesting natural carriers for data exfiltration.
  - It is possible to split data into several documents (of possible different formats)
- To some extent but in limited way (bytes to kilobytes of data), the same approach applies to network protocol metadata (see Drzymała & al. from Warsaw University of Technology for instance)

## Backdoors To Typical Case Complexity

Ryan Williams<sup>\*</sup>  
Computer Science Dept.  
Carnegie Mellon University  
Pittsburgh, PA 15213  
ryan@cs.cmu.edu

Carla P. Gomes<sup>†</sup>  
Dept. of Computer Science  
Cornell University  
Ithaca, NY 14853  
gomes@cs.cornell.edu

Bart Selman<sup>†</sup>  
Dept. of Computer Science  
Cornell University  
Ithaca, NY 14853  
selman@cs.cornell.edu

### Abstract

There has been significant recent progress in reasoning and constraint processing methods. In areas such as planning and finite model-checking, current solution techniques can handle combinatorial problems with up to a million variables and five million constraints. The good scaling behavior of these methods appears to defy what one would expect based on a worst-case complexity analysis. In order to bridge this gap between theory and practice, we propose a new framework for studying the

based planners, e.g. [11; 8; 1]. Somewhat surprisingly, on practical problem instances these methods scale well beyond what one might expect based on a formal complexity analysis. In fact, current state-of-the-art SAT solvers can handle problem instances, as they arise in finite model-checking and planning, with up to a million variables and five million clauses [15]. The success of these methods appears to hinge on a combination of two factors: (1) practical combinatorial problem instances generally have a substantial amount of (hidden) tractable sub-structure, and (2) new algorithmic techniques exploit such tractable structure, through, e.g., randomization and constraint learning.

- Existing PDF file after leak injection.
  - Lame example for illustration purposes.
  - Fully working PDF (no alert)
  - Leaked data not encoded (see further) for visibility purposes.
- In case of document integrity in place (rarely), documents may be created from scratch.

```
43 50@LSF@N0;?+T`f;cEiIm0Sf@OrsSbh`*"#S1>@JYB@-S100>-Xa&
44 Av09p;c0S0B0\X.S0B0`"vo007a5YN-BçEN7R,,G6Wt±jI001DZ=)I
45 0
46 :@N0|S1RSçXfRS0C4a0aXl0VX6H8+Y`çucpAzAY-ÂD0m\0%×`'£K-
47 g&k0G0C3su`>;0Y0V00-x0E0VMjB0G0-×t|N170S%4PH9RS+V0aZç
48 BSvX80M;AçK0qî&t`*gb<BSikAGlPz70j0C3#iZ0pWY0P+M80`çfE8G
49 yâRS`RSâi<cPy_0~8
50 Ec000aD00H09u00mçz000`i00RSi*q.T!;â0EÄ^4?;<I000ANjÄ0E0
51 endobj
52 ==Beginning of secret message+Abstract = The recent evol
53 - I1. Data may be analyzed so semantic detection (keyw
54 - I2. Encrypting data before exfiltration is likely to
55 - I3. Encryption means a secret key that can be recover
56 - I4. All outbound traffic may encrypted automatically
57
58 This talk intends to show how an attacker could exfiltrat
59
60 + End of secret message
61 5 0 obj
62 5671
63 endobj
64 3 0 obj
65 <<
```

# Entropy & Statistical Profiles

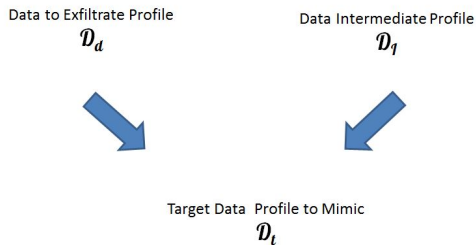
- First step of detection is generally automated. A statistical end entropy analysis may be enforced (even if most DLPs actually do not)
- So any data exfiltrated must exhibit innocent-looking profiles or at least compliant with the target environment.
- The general approach must
  - Consider key-dependent transformations of the data to be exfiltrated (otherwise it is encoding which can be detected and broken rather easily)
  - Prevent any manual analysis to reveal/recover the secret keys used.
  - Be able to mimic/simulate any target statistical/entropy profile
- In the rest of the talk, without loss of generality we focus on character entropy or statistical profile but any other profile can be considered ( $n$ -gram profiles)

# Secret Keys Management

- Whenever using key-dependent transformations (e.g. encryption), keys are the critical parts to protect from analysis.
- The solution is to use random keys generated from the environment (e.g. `/dev/random/` or equivalent)
  - For each file  $m$ , the malware generates a random key  $K_m$  of size  $|K|$ .
  - Size  $|K|$  must enable exhaustive search for the attacker. For instance, we tested  $|K| = 40$ .
- It is also possible to use encryption algorithms with backdoor to use longer key size (see my webpage for references).



# General Principle of Data Transformation



- Data to exfiltrate exhibit entropy profile  $H(\mathcal{D}_d)$  (e.g. \*.pdf files)
- Data to mimic exhibit entropy profile  $H(\mathcal{D}_t)$  (e.g. \*.text files).
- We then compute *a priori* an intermediate (transition) entropy profile  $H(\mathcal{D}_I)$  such that  $H(\mathcal{D}_t)$  is the joint entropy profile  $H(\mathcal{D}_d, \mathcal{D}_I)$  (here  $H(\mathcal{D})$  describes the entropy profile of source data with distribution  $\mathcal{D}$ ).

Without loss of generalities, let us consider the following text to evade

THIS IS A SECRET MESSAGE IN TEXT

The result of transformation toward different text distributions gives:

- Second-order Markov character distribution

HEMB THAT WILSDOM ABOARICE AMOLL ELETS XEDEAT GIRLS ESSE  
OFTE AGENT

- First-order Markov word distribution gives

ACTING THIS AND BEARING SECRET DEFENSE IS A NATURAL AGE IN  
METHOD of TEXT FOR THE LETTERS IN MESS BE THOSE

- There exist a quite infinite number of possibilities:
  - You can change the language (English to French for instance)
  - You can change the format (TEXT to WORD or PDF). However you may have to care about some tags to avoid errors.
  - You can make the distribution order vary...
- Note that the higher the distribution order the lower the data rate.
- Preventing this is impossible. It would require to recode/transcode any data before transmission preventively and by default (computing resources issues)
- No DLP tool is able to detect this nowadays.

# Summary of the talk

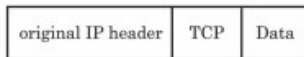
- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place**
- 6 Conclusion

- On sensitive networks, the key security goal is to forbid data wiretapping and eavesdropping.
- The most widespread solution is IPSec (or IPSec-like) tunnels (mandatory in military encrypting IP routers or IP encryptors [e.g. NATO]).
  - IPSec-based security is considered as the most efficient one.
- IPSec-based protocols can be manipulated to make data evade from “secure” computers.
  - Use of a covert channels (different protocols can be considered depending on the IPSEc implementation).
  - The technique is efficient even on complex traffics (multiplexed traffics, permanent or heavy traffics...).
  - Developed in C (server)/Rebol (client) in 2008, updated in 2021 (see our CanSecWest 2011 talk and paper [5])

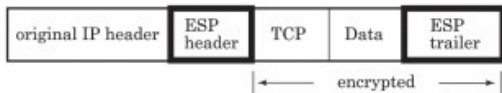
# ESP in Transport and Tunnel Mode

- Two sub-protocols:
  - AH : authentication and integrity.
  - ESP: AH + data encryption.
- Application-transparent security (*telnet, ftp, sendmail...*).

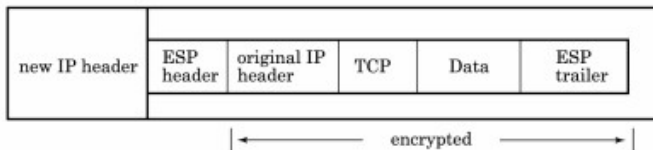
Before  
ESP



ESP IPv4  
Transport  
Mode

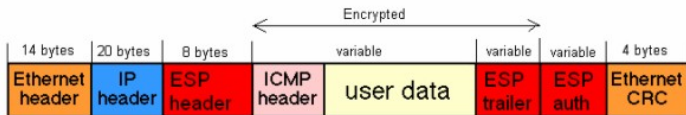
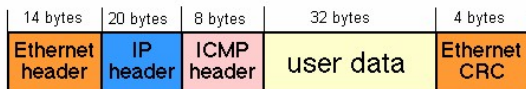


Tunnel  
Mode



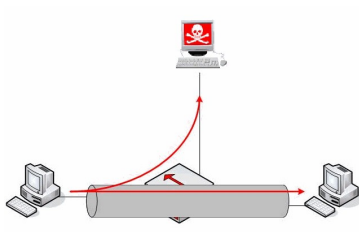
# ICMP (Ping) Packet

Our attack essentially considers ICMP (*ping*) packet with ESP encryption in tunnel mode (still exploitable in 2021 on many IPSec routers/software).



Other protocols and covert channels can also be used. But ICMP method is simple and illustrative enough for validation of the general concept.

# General Attack Scheme



- Alice and Bob communicate through an IPsec tunnel.
- The attacker wants to eavesdrop confidential data from Alice's computer. He can only observe the encrypted traffic and
  - Extract the IP header added by the IPsec device (e.g. a router in ESP tunnel mode) and get IP packets size.
- For other protocols, he only has to do such simple similar actions!
- Two-methods to exploit the covert-channel:
  - The *Ping Length method*.
  - The error-correcting codes-based optimized *Ping length method*.



# The *Ping Length* method

- One-to-one correspondence between data characters to evade and ICMP packet sizes.
- The attacker wiretaps the encrypted traffic and extracts the packet sizes to decode the data.
- Coding/decoding techniques must be powerful enough to cancel noise.
  - Data to evade are base-64 encoded (after transformation; see previous section).
  - Each character is repeated  $k$  times (5-repetition code in our case). The greater  $k$  the lower data rate.
  - Use of dedicated traffic tags: *Begin* and *Stop* tags.
  - To optimally manage the IPsec protocol (8-byte encryption), ping packet sizes must differ from at least eight units.

# Character Encoding

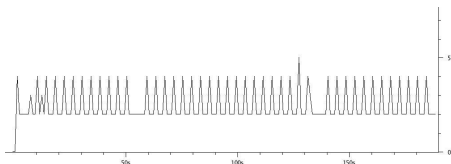
- Simple encoding ping packet size  $\leftrightarrow$  character value for text files (base64).
- 2021 version to adapt to partial padding techniques in a few IPSec routers.

## ping packet size $\leftrightarrow$ character value mapping

```
switch (length) {  
case 32: return '\t';  
case 96: return '\n';  
...  
case 288: return 'A';  
case 320: return 'B';  
case 352: return 'C';  
... }
```

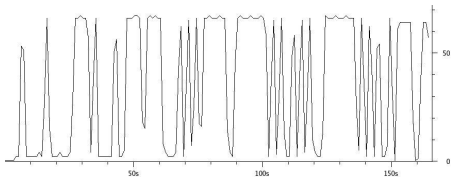
- Passively observes the packet flow and extracts suitable packets by using 5-repetition decoding techniques (ML decoding).
- Reverses the packet size/character mapping and decodes
- 5-repetition codes are powerful enough in most cases but noise reduction can be optimized by using suitable coding/decoding techniques (error-correcting codes-based optimized *Ping* length method).
- Let us present operational results when message “Salut comment ca va aujourd’hui ?” is emitted by the malware.
- Analysis: traffic load with respect to time.

# Experimental Results: Normal Traffic Load



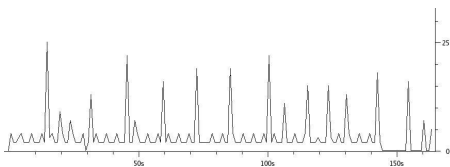
- No residual error.
- Total transmission time = 145 seconds.
- “Should” be easy to detect by good IDS (no TRANSEC).

# Experimental Results: Continuous Random Load (1Kb/s)



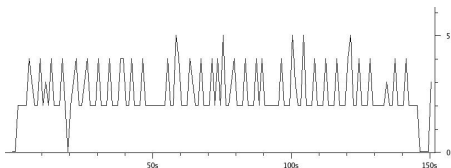
- Many errors (without decoding techniques).
- Total transmission time = 165 seconds.
- Can no longer be detected by IDS (traffic load hides malicious emission).
- Most usual cases (multi-user network).

# Experimental Results: 4 Kb/s Bursts with Random Phase



- A few errors (without decoding techniques).
- Total transmission time = 145 seconds.
- Can eventually be detected by IDS (weak TRANSEC).

# Experimental Results: Traffic With Random Bursts



- Two residual errors (“Salut commenB ca Aa aujourd’hui ?”) without error-correction.
- No transmission time increase.
- Difficult to detect with IDS.

- How to bypass IDS detection really?
- How to optimally correct residual decoding errors?
- Use heavily loaded traffics.
  - However, we have observed that on most real networks the traffic load is high enough to hide our malicious communication.
- To decode without residual errors, new coding/decoding schemes must be used.
- Use of more sophisticated data synchronisation/tagging techniques based on combinatorial patterns (needs more maths)
- Data are encoded under their hex value.



# Optimizations: Efficient Data Encoding

Efficient one-to-one character/size mapping (2021 version):

Character	0	1	2	3	4	5	6	7
Packet length	160	192	208	224	240	256	288	320
Character	8	9	A	B	C	D	E	F
Packet length	480	512	544	576	608	640	672	708

- Efficient at bypassing IPSec fragmentation effect. Packet size values are limited to a reduced interval (here [160, 708]).
- Use of n-repetition codes (among the most powerful error-correcting codes).

## Optimizations (2): $n$ -repetition Codes

In most traffics, packet sizes are uniformly distributed (however the attacker can perform a prior statistical analysis of the output traffic to recover the actual probability law).

Let us denote by  $p_i$  the probability of occurrence of a packet of size  $i$  (under the uniform law hypothesis  $p_i = \frac{1}{1514}$ ). In a “window” of  $p$  packets ( $n < p$ ),

- In normal conditions (e.g. without the malware) a (non necessary contiguous) pattern of  $n$  times the packet size  $s$  occurs in average  $\binom{p}{n} \cdot p_i^n$ .
- According to the traffic load (which has an impact on the window size  $p$ ) then choose the value  $n$  such that this probability is negligible.
- Experiments have shown that for most traffics  $n \in \{5, 7, 9, 11\}$  the residual decoding error probability tends towards 0.

- Other protocols than ICMP can be also used (DNS requests, HTTP requests, TTL, hop limit...).
- Detection with IDS is impossible (intractable to monitor all possible protocols/streams/methods especially for heavily loaded traffics).
- More sophisticated combinatorial coding/decoding techniques are possible to
  - To manage heavily loaded traffic with a large number of co-emitters.
  - Reduce the bandwidth consumption of the covert channel.
  - Reduce the network signature.
- Malware network-adaptative behaviours (to the traffic load for example).






# Summary of the talk

- 1 Introduction
- 2 State-of-the-Art & Technical Background
- 3 Non-Trivial Deniable Cryptography
- 4 Entropy and Statistical Profile Mimicking
- 5 Hijacking Encrypted Channels in Place
- 6 Conclusion**

- Exfiltrating data without detection (IDS, DLP, flow analysis...) is still easy.
- Detection faces complexity and computing issues especially if the malware embeds adaptive behaviors and techniques.
- The huge potential of malicious mathematics/cryptography is likely to see new malware technologies arise very soon (if not already for APTs).
- Prior security assessment, secure architecture, data assessment, tight rights management are necessary to lower the issues but in no way sufficient.
- The only solution would be to perform a systematic recoding/transcoding of outbound data.

Thank you for your attention  
Questions & Answers

# Bibliography

-  E. Filiol (2012). *Malicious Cryptography and Mathematics*.  
<https://www.intechopen.com/chapters/29700> (open access)
-  E. Carrera (2007). *Scanning data for entropy anomalies*.  
<http://blog.dkbza.org/2007/05/scanning-data-for-entropy-anomalies.html>
-  J. Fridrich (2010). *Steganography in Digital Media*. Cambridge University Press
-  F. Plumerault, B. David (2021). *DBI, debuggers, VM: gotta catch them all*, *Journal of Computer Virology and Hacking Techniques*, vol. 17, issue 2.
-  E. Filiol, F. Jennequin & G. Delaunay. “Malware-based Information Leakage over IPSEC Tunnels”, *Journal in Information Warfare*, volume 7, issue 3, pp. 11–22, December 2008.