

CONSTRUISONS **ENSEMBLE**
LA DÉFENSE DE DEMAIN

(Cyber-)Sécurité physique des systèmes embarqués

Benoît GÉRARD
DGA & IRISA



Qui suis-je ?

Diplômes

- Master d'algèbre appliquée.
- Doctorat en cryptologie.

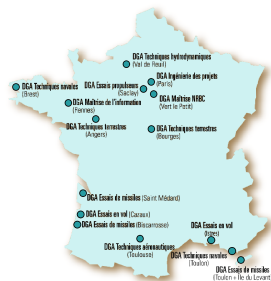
Activités professionnelles

- Ingénieur à la **DGA** (centre DGA-MI).
- Chercheur associé à l'**IRISA** (équipe EMSEC).
- Enseignant vacataires à l'UR1, INSA, Supélec.
- Maître d'apprentissage (apprentissage en 4ème année).

La DGA ?

Direction Générale de l'Armement

- Créée en 1961.
- Trois missions :
 1. équipement des forces armées,
 2. exportation de systèmes d'armement,
 3. développement de systèmes de défense de pointe.
- Quinze centres en France.



Et près de Rennes ...

DGA-MI ?

DGA Maîtrise de l'Information

- Centre créé en 1968.
- Plus de 1500 personnes y travaillent,
- dont une majorité en SSI.



DGA : qui travaille sur le sujet ?

Conception

- conception des algorithmes (cryptologie),
- conception des logiciels (gestion de projet, C),
- conception des composants (gestion de projet, HDL),
- conception des équipements (gestion de projet, modèles).

Évaluation

- évaluation non-invasive (électronique, cryptologie),
- évaluation invasive (électronique, physique, chimie),
- évaluation des équipements (interfaces de comm., techniques d'ouverture),
- évaluation tempest (traitement du signal, CEM).

L'IRISA ?

Institut de Recherche en Informatique et Systèmes Aléatoires

UMR (unité mixte de recherche)

- créée en 1975,
- plus de 850 personnes,
- Rennes, Lannion et Vannes,
- 7 axes
 - **la cybersécurité,**
 - la robotique,
 - l'énergie,
 - la santé,
 - l'environnement,
 - les transports,
 - la culture.

Tutelles

- CentraleSupélec,
- CNRS,
- ENS Rennes,
- IMT Atlantique,
- Inria,
- INSA Rennes,
- UBS,
- UR1.

IRISA : qui travaille sur le sujet ?

EMSEC *embedded security & cryptography*

- modèles de sécurité,
- conception de primitives/schémas/protocoles,
- sécurité des systèmes ubiquitaires.

CIDRE *confidentiality, integrity, disponibilité, repartition*

- compréhension des attaques,
- détection des attaques,
- résistance aux attaques.

TAMIS *threat analysis and mitigation for information sec.*

- analyse de malware,
- analyse de vulnérabilité (matérielle).

Introduction

Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Des applications et des contextes variés



Les systèmes embarqués

Contexte

Quelques spécificités rencontrées dans le monde embarqué.

Ressources limitées Contraintes

- puissance limitée,
- faible autonomie (énergie).
- surface limitée,
- contraintes temps réel,
- conditions environnementales peu clémentes (température, rayons ionisants).

Contexte d'emploi

- Souvent **facilement accessible à l'attaquant !**

Les systèmes embarqués

Accessibilité

Accessibilité logique

Les systèmes embarqués le sont dans un tout (e.g. carte SIM dans un smartphone).

L'attaquant peut accéder au système via ses interfaces avec ce tout.

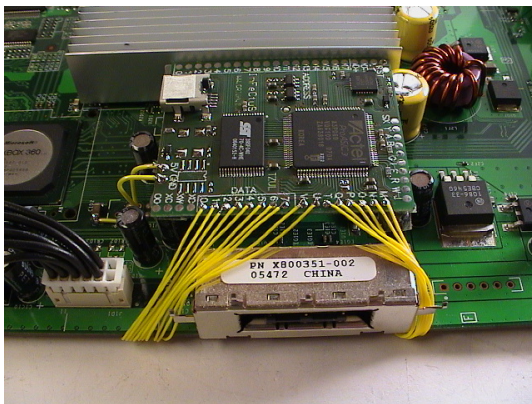
Accessibilité physique

Les systèmes embarqués se trouvent souvent dans des objets utilisés dans un environnement **non sûr**.

L'attaquant a donc souvent un accès physique au système embarqué.

Les systèmes embarqués

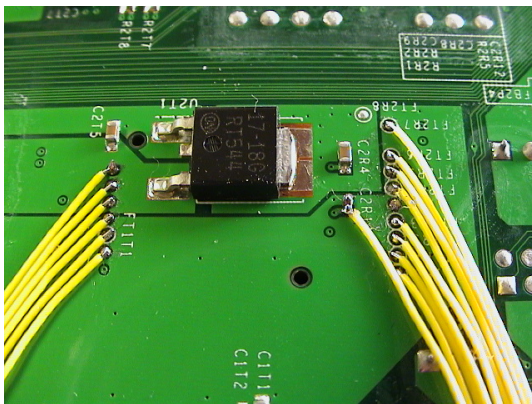
Exemple des consoles de jeux



Branchement d'un module à la mémoire nand (dessus).

Les systèmes embarqués

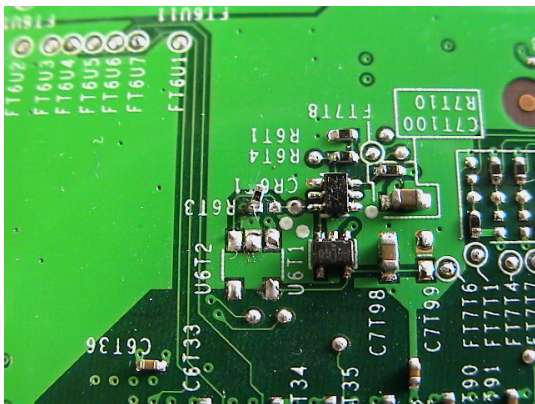
Exemple des consoles de jeux



Branchement d'un module à la mémoire nand (dessous).

Les systèmes embarqués

Exemple des consoles de jeux



Désactivation des efuses (grâce à la résistance).

Protections dans l'embarqué

Opacification

Il existe de nombreuses techniques complémentaires :

- manque de documentation,
- obfuscation de code,
- brouillage de mémoire (mélange des bits et/ou des adresses),
- enfouissement de pistes sur le PCB,
- utilisation de protocoles "maison" plutôt que de standards,
- ...

Elles permettent de ralentir les attaquants
mais cela ne suffit pas en général.

Protections dans l'embarqué

Utilisation de la cryptographie

Une autre piste est l'utilisation de **cryptographie** !

Les trois principaux services fournis par la cryptographie sont :

- la confidentialité,
- l'intégrité,
- l'authenticité.

La confidentialité permet de rendre bénin la lecture d'un bus ou d'une mémoire.

L'authenticité et l'intégrité permettent d'éviter les altérations d'un code critique.

La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



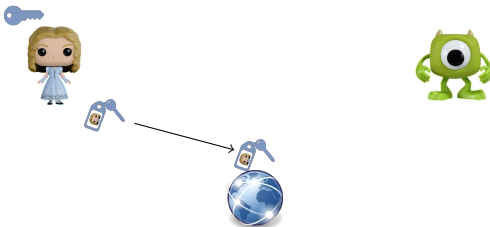
La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



La cryptographie

Symétrique vs asymétrique

Cryptographie symétrique



Cryptographie asymétrique



La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code
Chiffrement + MAC = chiffrement authentifié.

La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code
Chiffrement + MAC = chiffrement authentifié.

La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code

Chiffrement + MAC = chiffrement authentifié.

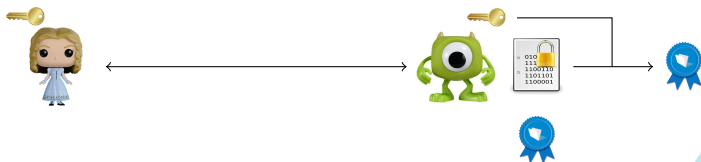
La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code

Chiffrement + MAC = chiffrement authentifié.

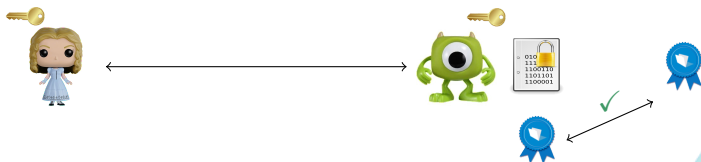
La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code

Chiffrement + MAC = chiffrement authentifié.

La cryptographie

Chiffrement et chiffrement authentifié

CHIFFREMENT ET MAC

Objectifs

- s'assurer de la **confidentialité** d'un contenu,
- s'assurer de l'**intégrité** d'un contenu (contenu non modifié par quelqu'un ne connaissant pas la clef).



MAC = Message Authentication Code

Chiffrement + MAC = chiffrement authentifié.

La cryptographie

Signature

SIGNATURE

Objectifs

- s'assurer de l'**intégrité** d'un contenu,
- s'assurer de l'**authenticité** (preuve de l'origine du contenu).



La cryptographie

Signature

SIGNATURE

Objectifs

- s'assurer de l'**intégrité** d'un contenu,
- s'assurer de l'**authenticité** (preuve de l'origine du contenu).



La cryptographie

Signature

SIGNATURE

Objectifs

- s'assurer de l'**intégrité** d'un contenu,
- s'assurer de l'**authenticité** (preuve de l'origine du contenu).



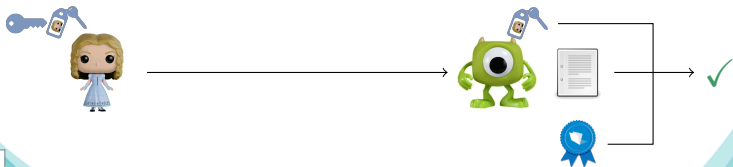
La cryptographie

Signature

SIGNATURE

Objectifs

- s'assurer de l'**intégrité** d'un contenu,
- s'assurer de l'**authenticité** (preuve de l'origine du contenu).



La cryptographie

Chiffrement asymétrique et échange de clefs

Cryptographie symétrique \Rightarrow une clef par correspondant !

CHIFFREMENT ASYMÉTRIQUE

- Utilise le même principe de paire de clefs que la signature,
- le secret est du côté de la personne qui déchiffre,
- les opérations sont coûteuses.

ÉCHANGE DE CLEF

- Secret partagé grâce à des opérations coûteuses,
- pour ensuite utiliser la cryptographie symétrique.

La cryptographie

L'homme du milieu

MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

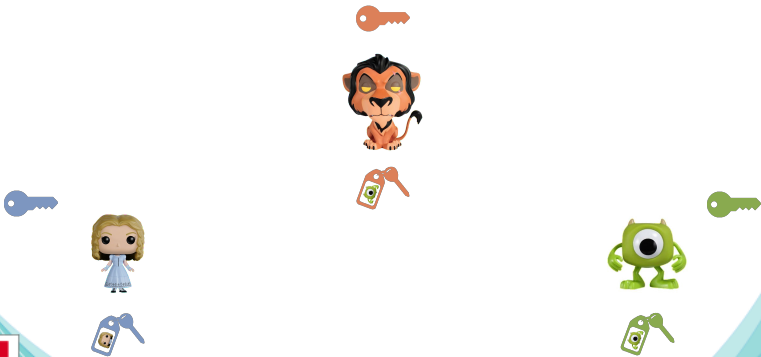
MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

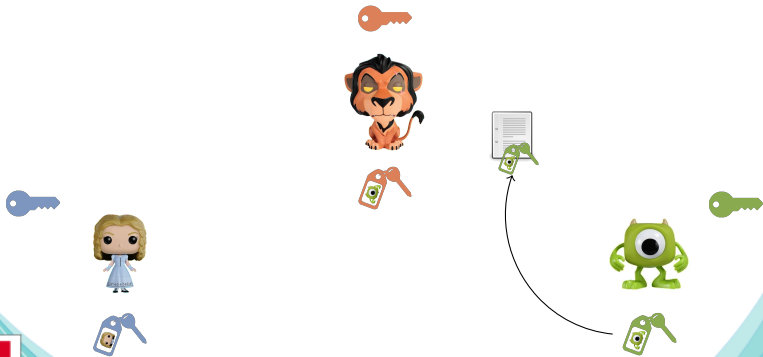
MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

MAN IN THE MIDDLE



La cryptographie

L'homme du milieu

MAN IN THE MIDDLE



La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



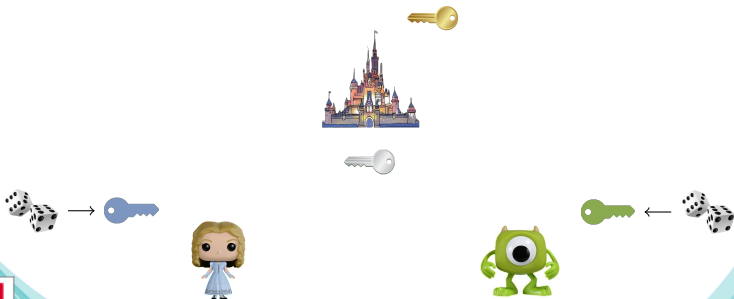
La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



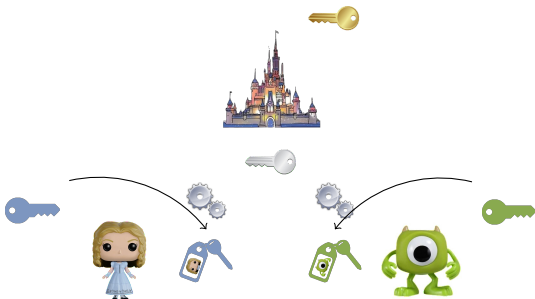
La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



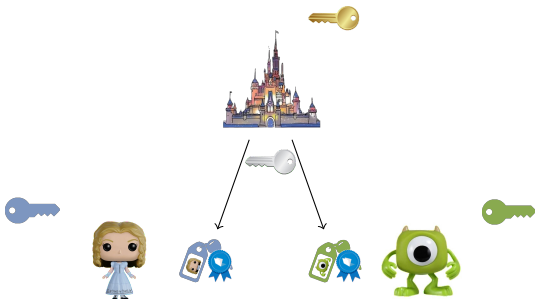
La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



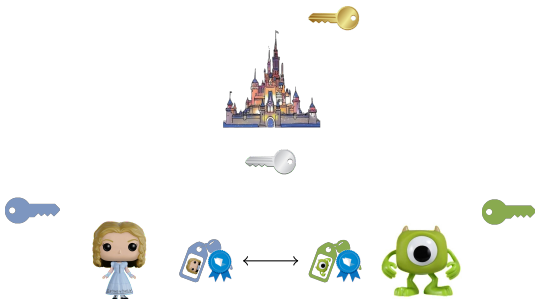
La cryptographie

Certificats

CERTIFICATS

Objectif

- s'assurer de l'**identité** de son correspondant.



Produits de sécurité

Produits de sécurité

Sécurité et ...sécurité

Sécurité : "Safety"

C'est la **sécurité des personnes** (e.g. avionique).

Sécurité : "Security"

C'est la **sécurité des données** (e.g. données classifiées).

En cas de coupure électrique, les portes d'entrée du bâtiment doivent :

- s'ouvrir pour laisser sortir (*safety*),
- se verrouiller pour éviter les espions (*security*).

Produits de sécurité

Sécurité et ...sécurité

Sécurité : "Safety"

C'est la **sécurité des personnes** (e.g. avionique).

Sécurité : "Security" ← cet exposé

C'est la **sécurité des données** (e.g. données classifiées).

En cas de coupure électrique, les portes d'entrée du bâtiment doivent :

- s'ouvrir pour laisser sortir (*safety*),
- se verrouiller pour éviter les espions (*security*).

Produits de sécurité

Lien avec l'embarqué

Un produit de sécurité :

- est un dispositif spécialisé dans la sécurité,
- qui effectue souvent ses tâches en autonomie,
- est inclus comme sous-système d'un plus gros système.

Un produit de sécurité présente donc les caractéristique d'un **système embarqué.**

Produits de sécurité

Quelques exemples



Disque chiffrant



HSM (Hardware Security Module)



Portefeuille électronique



Application bancaire
(sur smartphone)

Produits de sécurité

Problématiques industrielles

Besoins de sécurité

- Protection de la propriété intellectuelle.
- Protection des secrets du client.

Contraintes industrielles

- Coûts (temps, ressources humaines).
⇒ produits sur étagère, sous-traitance ...
- Flexibilité et adaptabilité au besoin client.
⇒ configurabilité, compatibilité ...

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?

- processeur, sous-traitant, employés . . .

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?
2. Quels sont les risques sur les éléments non fiables ?
 - perte de propriété intellectuelle, piégeage du produit ...

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?
2. Quels sont les risques sur les éléments non fiables ?
3. Quels sont les coûts des risques ?
 - perte de compétitivité, de marchés ...

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?
2. Quels sont les risques sur les éléments non fiables ?
3. Quels sont les coûts des risques ?
4. Quelles sont les probabilités que les risques se réalisent ?

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?
2. Quels sont les risques sur les éléments non fiables ?
3. Quels sont les coûts des risques ?
4. Quelles sont les probabilités que les risques se réalisent ?
5. Quels sont les coûts d'une protection ?

Produits de sécurité

Risques, confiance et coûts

1. En quoi/qui ai-je confiance ?
2. Quels sont les risques sur les éléments non fiables ?
3. Quels sont les coûts des risques ?
4. Quelles sont les probabilités que les risques se réalisent ?
5. Quels sont les coûts d'une protection ?
6. Jusqu'à quel niveau d'attaquant mon produit doit-il tenir ?
 - tant pis pour la NSA ...

Produits de sécurité

Évaluations

Critères Communs

« Évaluer de façon impartiale la sécurité des systèmes et des logiciels informatiques »

- 17 pays signataires + 9 acceptent les certificats,
- autorités de certification (ANSSI, BSI, NIAP ...),
- laboratoires accrédités (CESTI/ITSEF, CCTL ...),
- niveaux de 1 à 7 + CSPN.

Autres schémas

- Europay, MasterCard, VISA, EMVCo,
- Global Platform,
- ...

Produits de sécurité

Cybersécurité physique



Produits de sécurité

Cybersécurité physique



attaques actives

Produits de sécurité

Cybersécurité physique



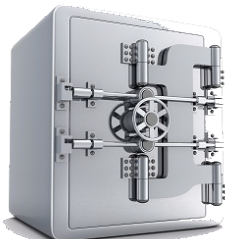
attaques actives



attaques passives

Produits de sécurité

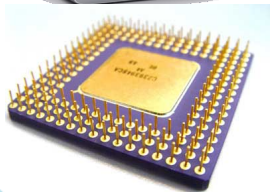
Cybersécurité physique



attaques actives

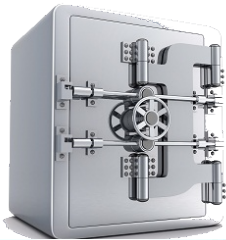


attaques passives



Produits de sécurité

Cybersécurité physique



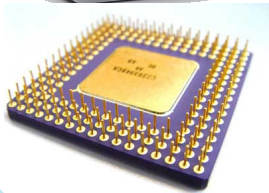
attaques actives



attaques passives

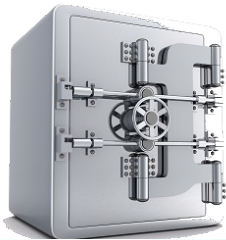


attaques actives



Produits de sécurité

Cybersécurité physique



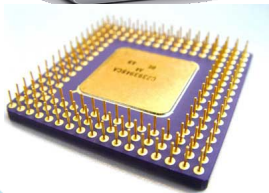
attaques actives



attaques passives



attaques actives



attaques passives

Produits de sécurité

Évaluation de la sécurité physique

Si risque d'attaques physiques : laboratoires spécialisés.

- 3 CESTIs « matériel » en France.

Score pour chaque attaque en fonction :

- du temps passé,
- du niveau d'expertise requis,
- la connaissance du produit,
- le nombre d'échantillons nécessaires,
- l'équipement requis pour mener l'attaque.

Attaques par canaux auxiliaires

Attaques par canaux auxiliaires

Présentation globale

Auxiliaire \Rightarrow plusieurs canaux ?

Lors d'une discussion de visu :

- canal principal : audio (mots utilisés, intonation),
- autre canal : vidéo (expressions du visage, gestuelle).

Via ce second canal, on peut :

1. préciser le message (schéma explicatif),
2. faire passer un message contradictoire (clin d'œil),
3. confondre les sentiments réels de l'orateur (mensonge).

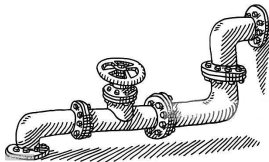


Canal auxiliaire vs canal caché

Canal auxiliaire



Canal caché



Canal auxiliaire vs canal caché

Canal auxiliaire

- Mettre en défaut les propriétés de sécurité telles que
 - la confidentialité (des secrets, du savoir-faire),
 - l'authenticité (des condensats de mots de passe, des mises-à-jour).

Canal caché

- Mettre en défaut les propriétés d'isolation
 - entre un PC confidentiel et un PC connecté à internet (air gap),
 - entre deux processus s'exécutant sur un même processeur (cloud).

Quels canaux ?

Différents types de canaux utilisables :

- temps d'exécution,
- consommation de courant,
- rayonnement électromagnétique,
- son émis,
- température,
- émission de lumière,
- ...

Temps de vérification d'un PIN

Exemple simplifié et didactique de Joe Grand.

<https://youtu.be/2-zQp26nbY8>

Temps de calcul (signature ECDSA)

Type de vulnérabilité connue depuis . . .15 ans !

signature | vulnérabilité | technologie

Trusted Platform Module (TPM)-Fail – two new CPU vulnerabilities allow attackers to retrieve cryptographic keys

By admin - November 13, 2019

310



VULNÉRABILITÉS | CYBERATTQUES | MALWARES | DIVERS | BRÈCHE DE DONNÉES



Vulnérabilités

TPM-Fail, 2 vulnérabilités affectant des milliards d'appareils

Minerva attack can recover private keys from smart cards, cryptographic libraries

Older Athena IDProtect smart cards are impacted, along with the WolfSSL, MatrixSSL, Crypto++, Oracle SunEC, and Libgcrypt crypto libraries.

By Clément Chevillon for SecWiki - October 8, 2019 - 0:00:00



MORE FROM LETSUNA COUNCIL

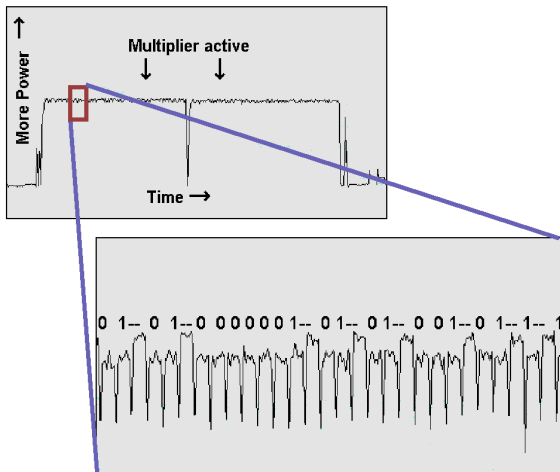
Security: OWASP MAS projects targeted in another set of zero-days attacks

Security: Chrome, Java, and React patched together to end out-of-band exploits

Security: Google Chrome and Service Fabric targeted by Baku and Tropic campaigns

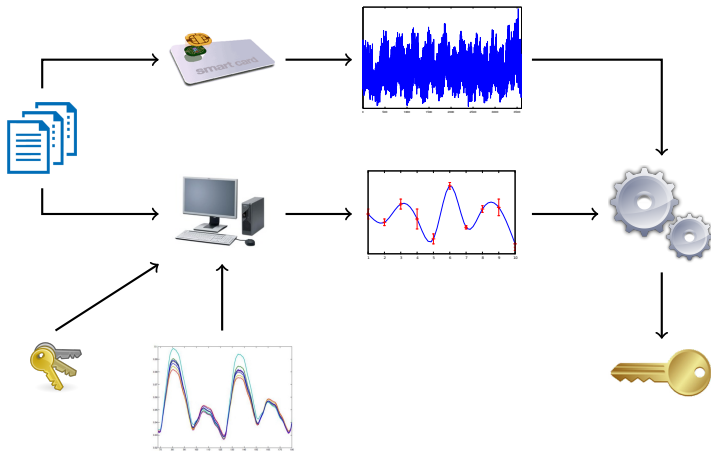
Analyse de courant

Cryptographie asymétrique



Analyse de courant

Cryptographie symétrique



Rayonnement électromagnétique

Similaire aux attaques par analyse de courant.

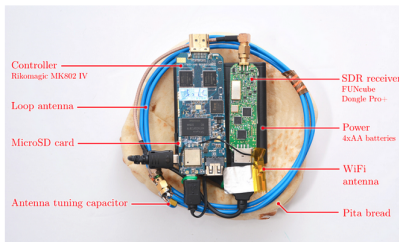


PITA Side-Channel Attack Steals GPG Key from Laptops

Author:
Michael Mimoso
June 24, 2015 / 11:27 am
2:30 minute read

Share this article:

f t ...



Researchers at Tel Aviv University have developed a compact, untethered tool capable of extracting GnuPG crypto keys (RSA and

INFOSEC INSIDER

Understanding
Email Attacks
Team

June 4, 2020

Long Tail Anal
Cybercrime B

May 21, 2020

The Windows
at Stake

May 19, 2020

VPN Concern
Remote Empl

May 5, 2020

Building for B
Security Conc
Scale

April 30, 2020

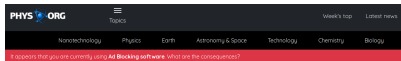


Sons divers

le Reports Security Social Networks Terrorism ICS-SC

“We observe that GnuPG’s RSA signing (or decryption) operations are readily identified by their acoustic frequency spectrum. Moreover, the spectrum is often key-dependent, so that secret keys can be distinguished by the sound made when they are used. The same applies to ElGamal decryption.”

The researchers observed that the acoustic attack range surpassed 4 meters using a sensitive parabolic microphone, meanwhile without this kind of receiver they achieved a range of 1 meter.



Research trio crack RSA encryption keys by listening to computer noise



UC Berkeley News

NewsCenter

Today's news & events

Subscribe to news

For the news media

Calendar of events

Search Berkeley News

Press Release

Researchers recover typed text using audio recording of keystrokes

By Sarah Yang, Media Relations | 14 September 2005

BERKELEY – A new security threat revealed by computer scientists at the University of California, Berkeley, may be enough to drive some people away from their computer

Share



Clever Attack Uses the Sound of a Computer's Fan to Steal Data



Et même la vidéo !

INDUSTRY / MOBILE

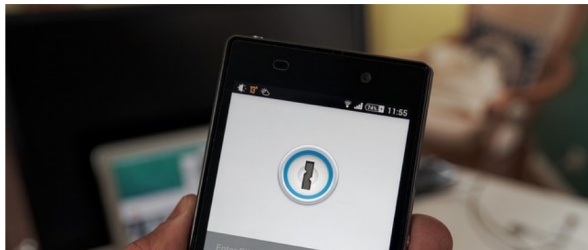
Hackers May Be Able to Use the Sensors on Your Phone to Guess Your Pin

Using the power of deep learning and the six sensors on smartphones, researchers were able to hack into an android phone by guessing the pin correctly.



By Donovan Alexander

December 29, 2017



En pratique, pas toujours évident

En fonction des contextes la mise en œuvre de ces attaques est plus ou moins évidente.

<https://youtu.be/FktI4qSjzaE?t=518>

Mais cela ne fera que ralentir un (groupe d') attaquant(s) motivé(s) et compétent(s).

Attaques par canaux auxiliaires

Quelques exemples de contre-mesures

Cryptographie asymétrique

Simple Power Analysis

EXEMPLE DU RSA

Calcul de $p = c \bmod N$

```
r = 1
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
```

01 1 001

↓

SSMSMSSSM

01 1 001

return r

Square and Multiply

Cryptographie asymétrique

Algorithme régulier

EXEMPLE DU RSA

Calcul de $p = c \text{ mod } N$

```
r = 1
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
  else
    t = r * c mod N
return r
```

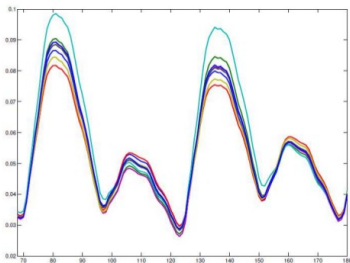
0 1 1 0 0 1
↓
SMSMSMSMSM
? ? ? ? ? ?

Square and Multiply Always

Cryptographie symétrique

Modèle de fuite #1

On ne regarde plus des motifs mais les variations d'amplitude du signal.



Cryptographie symétrique

Modèle de fuite #2

Modèle poids de Hamming (*HW*)

La fuite dépend du nombre de 1 dans les données manipulées.

Quand on envoie un signal

- 0 pas de tension,
- 1 tension maintenue.

Modèle distance de Hamming (*HD*)

La fuite dépend de la différence entre les données.

Quand on met à jour une valeur

- ≠ modifier un bit induit une sur-consommation,
- = ne rien faire n'induit pas de sur-consommation.

Cryptographie symétrique

Differential Power Analysis

$$\left. \begin{array}{l} C = 0x7 \\ HW(C \oplus K) = 1 \end{array} \right\} \implies K \in \{0x3, 0x5, 0x6, 0xF\}$$

Cryptographie symétrique

Differential Power Analysis

$$\left. \begin{array}{l} C = 0x7 \\ HW(C \oplus K) = 1 \end{array} \right\} \implies K \in \{0x3, 0x5, 0x6, 0xF\}$$

$$\left. \begin{array}{l} C = 0x1 \\ HW(C \oplus K) = 3 \end{array} \right\} \implies K \in \{0x6, 0xA, 0xC, 0xF\}$$

Cryptographie symétrique

Differential Power Analysis

$$\left. \begin{array}{l} C = 0x7 \\ HW(C \oplus K) = 1 \end{array} \right\} \Rightarrow K \in \{0x3, 0x5, 0x6, 0xF\}$$

\cap

$$\left. \begin{array}{l} C = 0x1 \\ HW(C \oplus K) = 3 \end{array} \right\} \Rightarrow K \in \{0x6, 0xA, 0xC, 0xF\}$$

\downarrow

$\{0x6, 0xF\}$

Cryptographie symétrique

Pourquoi l'attaque fonctionne ?

La fuite/conso mmation (HW ou HD) dépend :

- du secret fixe : K ,
- d'une valeur connue de l'attaquant : C .

On peut donc :

- retrouver les K compatibles avec les valeurs de C et de $HW(C \oplus K)$,
- faire varier C pour cribler les valeurs de K restantes car K est fixe !

C'est le mécanisme du *Qui est-ce !*

Cryptographie symétrique

Concept du masquage

Objectif

Cacher les valeurs intermédiaires du calcul à l'attaquant.
En particulier $Z = C \oplus K$.

On va tirer R aléatoirement dans le composant et faire :

$$C' = C \oplus R$$

$$Z' = C' \oplus K$$

⋮

⋮

⋮

$$P = P' \oplus R$$

calculs masqués

Cryptographie symétrique

Concept du masquage

Objectif

Cacher les valeurs intermédiaires du calcul à l'attaquant.
En particulier $Z = C \oplus K$.

On va tirer R aléatoirement dans le composant et faire :

$$C' = C \oplus R \quad \rightsquigarrow \text{connu} \oplus \text{aléa}$$

$$Z' = C' \oplus K \quad \rightsquigarrow \text{aléa} \oplus \text{secret}$$

⋮

⋮

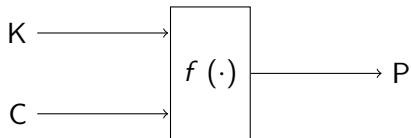
calculs masqués

⋮

$$P = P' \oplus R \quad \rightsquigarrow \text{aléa} \oplus \text{aléa}$$

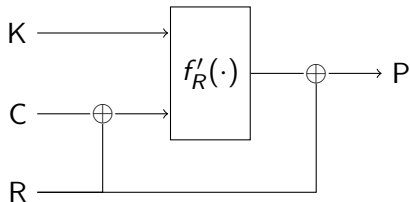
Cryptographie symétrique

Masquage d'une fonction #1



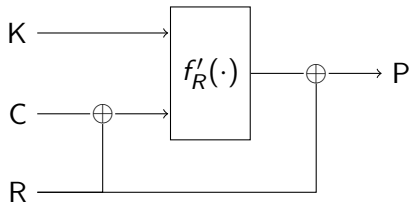
Cryptographie symétrique

Masquage d'une fonction #1



Cryptographie symétrique

Masquage d'une fonction #1

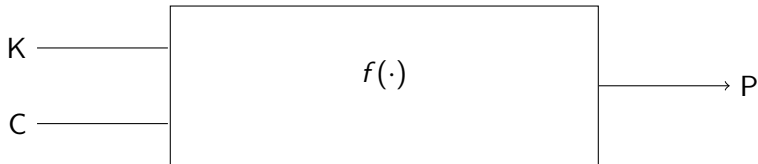


Disclaimer

Comme toute simplification elle n'est pas totalement correcte mais l'idée est là !

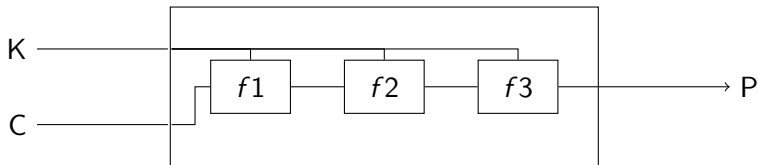
Cryptographie symétrique

Masquage d'une fonction #2



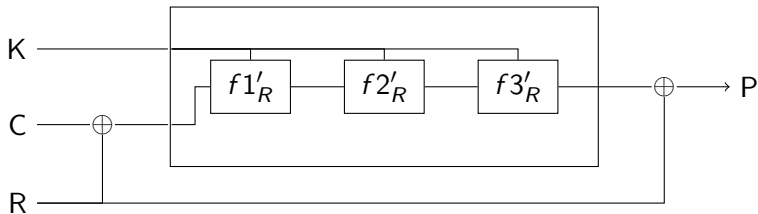
Cryptographie symétrique

Masquage d'une fonction #2



Cryptographie symétrique

Masquage d'une fonction #2



Il faut donc que

$$f'_R(X \oplus R) = f(X) \oplus R$$

Cryptographie symétrique

Masquage "facile" des blocs de base

Plus généralement

Il faut donc que

$$f'_R(X \star R) = f_i(X) \star R$$

type	$f(x, k)$	\longrightarrow	$f'_r(x \star r, k)$
\oplus	$x \oplus k$		$(x \oplus r) \oplus k = (x \oplus k) \oplus r$
$+$	$x + k$		$(x + r) + k = (x + k) + r$
\times	$x \times k$		$(x \times r) \times k = (x \times k) \times r$

Cryptographie symétrique

Masquage "facile" des blocs de base

Plus généralement

Il faut donc que

$$f'_R(X \star R) = f_i(X) \star R$$

type	$f(x, k)$	\longrightarrow	$f'_r(x \star r, k)$
\oplus	$x \oplus k$		$(x \oplus r) \oplus k = (x \oplus k) \oplus r$
$+$	$x + k$		$(x + r) + k = (x + k) + r$
\times	$x \times k$		$(x \times r) \times k = (x \times k) \times r$

Par contre

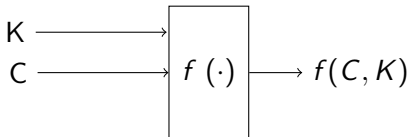
$$(x \oplus r) + k \neq (x + k) \oplus r$$

Cryptographie symétrique

Masquage : les autres cas

Dans les autres cas il faut se débrouiller !

- Tabulation des fonctions f'_R ,
- Découpage des calculs et utilisation de plus d'aléa.



Objectif

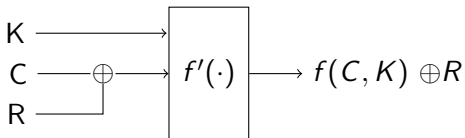
À chaque étape du calcul la valeur manipulée est statistiquement indépendante de K .

Cryptographie symétrique

Masquage : les autres cas

Dans les autres cas il faut se débrouiller !

- Tabulation des fonctions f'_R ,
- Découpage des calculs et utilisation de plus d'aléa.



Objectif

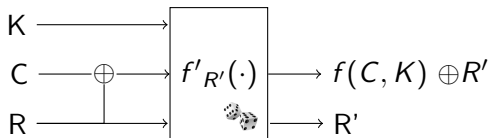
À chaque étape du calcul la valeur manipulée est statistiquement indépendante de K .

Cryptographie symétrique

Masquage : les autres cas

Dans les autres cas il faut se débrouiller !

- Tabulation des fonctions f'_R ,
- Découpage des calculs et utilisation de plus d'aléa.



Objectif

À chaque étape du calcul la valeur manipulée est statistiquement indépendante de K .

Compilation et sécurité

Compilation

Objectifs du compilateur

- Transformation code humain / binaire.

Compilation

Objectifs du compilateur

- Transformation code humain / binaire.
- Optimisations :
 - temps de calcul,
 - taille de code.

Compilation

Objectifs du compilateur

- Transformation code humain / binaire.
- Optimisations :
 - temps de calcul,
 - taille de code.

Pour cela :

1. analyse (lexicale, syntaxique, sémantique),
2. représentation(s) intermédiaires,
3. optimisations,
4. génération du binaire.

Quelques optimisations #1

Dead Store Elimination

Une écriture dans une variable qui

soit n'est pas relue ensuite,

soit est écrasée avant sa prochaine lecture,

peut être supprimée.

Square & Multiply Always

```
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
  else
    t = r * c mod N
return r
```

Quelques optimisations #1

Dead Store Elimination

Une écriture dans une variable qui

soit n'est pas relue ensuite,

soit est écrasée avant sa prochaine lecture,

peut être supprimée.

Square & Multiply Always

```
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
  else
    t = r * c mod N
return r
```

Quelques optimisations #1.5

Dead Store Elimination

Une écriture dans une variable qui

soit n'est pas relue ensuite,

soit est écrasée avant sa prochaine lecture,
peut être supprimée.

Effacement sécurisé

En fin de fonction effacer les valeurs temporaires en mémoire :

- valeur intermédiaire de calcul,
- clef, mot de passe . . .

Quelques optimisations #2

Algebraic simplification

On peut simplifier les expressions algébriques.

Square & Multiply Always

On pourrait éviter l'optimisation « dead store »

```
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
  else
    r = r * 1 mod N
return r
```

Quelques optimisations #2

Algebraic simplification

On peut simplifier les expressions algébriques.

Square & Multiply Always

On pourrait éviter l'optimisation « dead store »

```
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1
    r = r * c mod N
  else
    r = r * 1 mod N
return r
```


Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

$$reg_2 \leftarrow reg_2 \oplus reg_1$$

Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

$$reg_2 \leftarrow reg_2 \oplus reg_1 \rightsquigarrow HD(c, r \oplus c) = HW(r)$$

Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

$$reg_2 \leftarrow reg_2 \oplus reg_1 \rightsquigarrow HD(c, r \oplus c) = HW(r)$$

$$reg_1 \leftarrow reg_2 \oplus reg_3$$

Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

$$reg_2 \leftarrow reg_2 \oplus reg_1 \rightsquigarrow HD(c, r \oplus c) = HW(r)$$

$$reg_1 \leftarrow reg_2 \oplus reg_3 \rightsquigarrow HD(r, r \oplus c \oplus k) = HW(c \oplus k)$$

Quelques optimisations #3

Register Allocation

Le compilateur décide dans quels registres il stocke les variables.

Masquage et distance de Hamming

- But : xorer k à c sans fuiter $c \oplus k$.
- Masquage : $c' \leftarrow r \oplus c$ puis $z' \leftarrow c' \oplus k \dots$
- État initial : $reg_1 = r, reg_2 = c, reg_3 = k$.

$$reg_2 \leftarrow reg_2 \oplus reg_1 \rightsquigarrow HD(c, r \oplus c) = HW(r)$$

$$reg_1 \leftarrow reg_2 \oplus reg_3 \rightsquigarrow HD(r, r \oplus c \oplus k) = HW(c \oplus k)$$

Si il avait choisi reg_2 on aurait eu
 $HD(r \oplus c, r \oplus c \oplus k) = HW(k)$

Quelques optimisations #5

Attaques en fautes

Lors d'une attaque en faute l'attaquant tire profit d'une erreur qu'il arrive à provoquer durant le calcul.

On se protège en effectuant plusieurs fois le calcul et en comparant.

Common Subexpression Elimination

On peut factoriser le calcul de sous-expressions communes à plusieurs expressions.

En particulier si on calcule trois fois la même chose ...

Micro-architecture

Mécanismes de micro-architecture

Quelques exemples

Pipeline

- Instructions exécutées en plusieurs coups d'horloge.
- À chaque coup on commence le traitement d'une nouvelle instruction et on avance d'une étape pour les autres.

Exécution spéculative

Pour remplir le pipeline on commence à traiter une branche d'un if en espérant que ce soit la bonne.

Exécution hors d'ordre

On s'autorise à modifier l'ordre des opérations.

Exécution spéculative/hors d'ordre



ACTUALITÉS ▾ TESTS ET DOSSIERS ▾ PROMOS ET BONS PLANS ▾ GINJFO TV ▾ GAMING ▾

🏠 Accueil / Actualités / securite-informatique / Failles Meltdown et Spectre, votre PC Linux est-il vulnérable ?



Failles Meltdown et Spectre, votre PC Linux est-il vulnérable ?

Exécution spéculative

Branch Prediction et Square & Multiply Always #1

Square & Multiply Always

```
for i = k-1 to 0 do
  r = r * r mod N
  if d_i == 1          // branchement conditionnel
    r = r * c mod N
  else
    r = r * 1 mod N
return r
```

Le processeur va tenter de prédire le branchement

Exécution spéculative

Branch Prediction et Square & Multiply Always #2

Différentes techniques en fonction :

- de la connaissance (ou non) de l'algorithme de prédiction,
- de l'accès à des compteurs de performances,
- de la possibilité de rejouer plusieurs fois le calcul ciblé,
- de la possibilité de se synchroniser au programme ciblé
- ...

Attention

Pour RSA : même s'il manque quelques bits de la clef
on peut la reconstruire !

Pipeline et Masquage

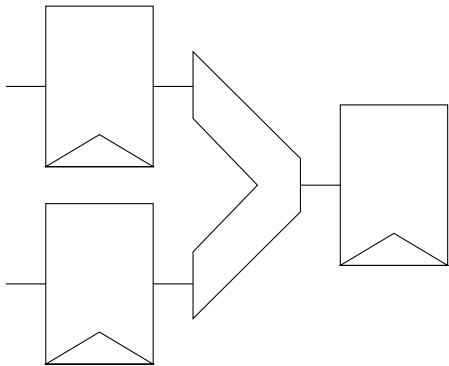


Schéma de l'ALU

État des registres

<i>r1</i>	<i>C</i>
<i>r2</i>	<i>R</i>
<i>r3</i>	0

Instructions

Pipeline et Masquage

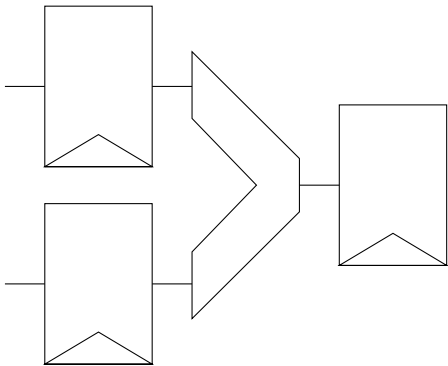


Schéma de l'ALU

État des registres

<i>r1</i>	<i>C</i>
<i>r2</i>	<i>R</i>
<i>r3</i>	0

Instructions

$r3 \leftarrow r1 \oplus r2$

Pipeline et Masquage

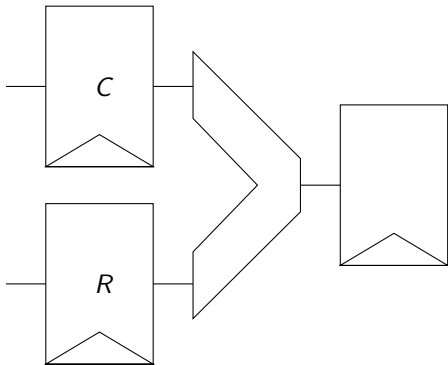


Schéma de l'ALU

État des registres

<i>r1</i>	<i>C</i>
<i>r2</i>	<i>R</i>
<i>r3</i>	0

Instructions

$r3 \leftarrow r1 \oplus r2$

Pipeline et Masquage

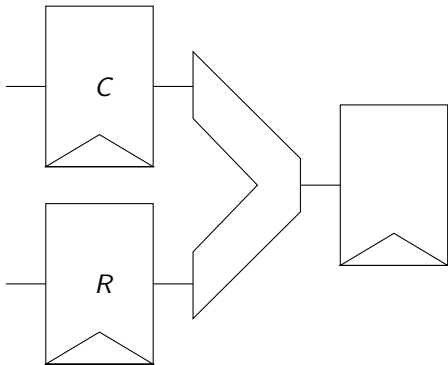


Schéma de l'ALU

État des registres

r1	C
r2	R
r3	0

Instructions

r3	\leftarrow	r1	\oplus	r2
r2	\leftarrow	ld	@K	

Pipeline et Masquage

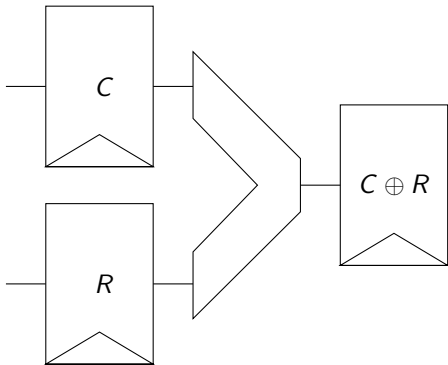


Schéma de l'ALU

État des registres

r1	C
r2	K
r3	0

Instructions

r3	\leftarrow	r1	\oplus	r2
r2	\leftarrow	ld	@K	

Pipeline et Masquage

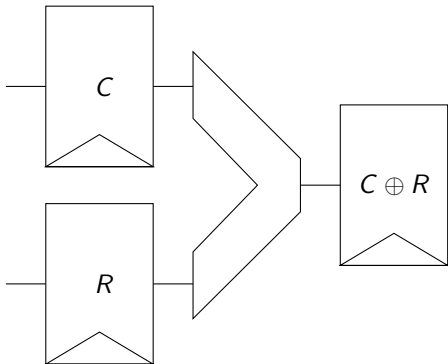


Schéma de l'ALU

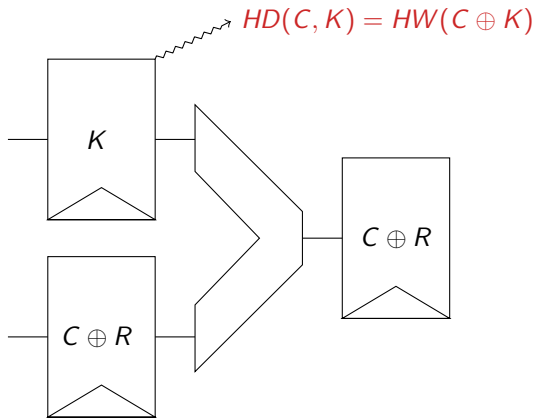
État des registres

$r1$	C
$r2$	K
$r3$	$C \oplus R$

Instructions

$r3$	\leftarrow	$r1 \oplus r2$
$r2$	\leftarrow	$ld @K$
$r1$	\leftarrow	$r2 \oplus r3$

Pipeline et Masquage



État des registres

$r1$	C
$r2$	K
$r3$	$C \oplus R$

Instructions

$r3$	\leftarrow	$r1 \oplus r2$
$r2$	\leftarrow	$ld @K$
$r1$	\leftarrow	$r2 \oplus r3$

Conclusion

Conclusion

La sécurité informatique





- Jeu du chat et de la souris attaque/défense.
- Des besoins à l'opposé des outils classiques.

La « cybersécurité physique »

- A de beaux jours à venir avec les objets connectés.
- Domaine de pointe pluridisciplinaire !.

Des questions ?

Références

-  Onur Aciicmez, Çetin Kaya Koç, and Jean-Pierre Seifert, Predicting secret keys via branch prediction, CT-RSA 2007, pp. 225–242.
-  David Berend, Shivam Bhasin, and Bernhard Jungk, There goes your PIN : exploiting smartphone sensor fusion under single and cross user setting, ARES 2018, pp. 54 :1–54 :10.
-  Billy Bob Brumley and Nicola Taveri, Remote timing attacks are still practical, ESORICS 2011, pp. 355–371.
-  Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer, Stealing keys from pcs using a radio : Cheap electromagnetic attacks on windowed exponentiation, CHES 2015, pp. 207–228.

Références

-  Daniel Genkin, Adi Shamir, and Eran Tromer, Acoustic cryptanalysis, J. Cryptology **30** (2017), no. 2, 392–443.
-  Paul Kocher, Jann Horn, Anders Fogh, , Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom, Spectre attacks : Exploiting speculative execution, S&P'19.
-  Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, Differential power analysis, CRYPTO '99, pp. 388–397.
-  Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg, Meltdown : Reading kernel memory from user space, USENIX Security 18.

Références



Minerva website,

<https://minerva.crocs.fi.muni.cz/>.



Daniel Moghimi, Berk Sunar, Thomas Eisenbarth, and
Nadia Heninger,

TPM-FAIL : TPM meets Timing and Lattice Attacks,
USENIX Security 20.