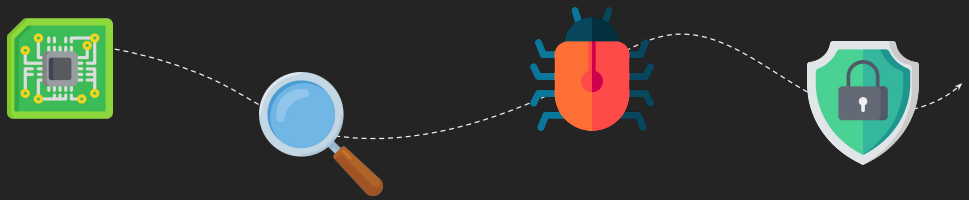
Cybersécurité à la loupe : du logiciel au matériel



Clémentine Maurice, chargée de recherche CNRS
Laboratoire CRIStAL, équipe-projet Inria Spirals
01/10/2025 - Bar des Sciences, Université de Lille

PARCOURS

2007-2012
Double diplôme ingénieur informatique et master recherche, INSA Rennes

2012-2015
Thèse CIFRE, Technicolor/EURECOM

2016-2017
Postdoctorat, Graz University of Technology, Autriche

depuis 2017
Chercheuse au CNRS

- 2017-2021 : à l'IRISA, Rennes
- depuis 2021 : à CRIStAL, Lille

La ville de Lille victime d'une cyberattaque, quatre agents municipaux ont reçu une demande de rançon

L'attaque informatique, qui empêche notamment l'accès au standard de la municipalité, est en cours depuis le 28 février.

franceinfo
Radio France

Publié le 15/03/2023 22:16 | Temps de lecture : 1 min

Paris 2024 : 548 signalements et incidents de cybersécurité liés aux Jeux décomptés entre mai et septembre

D'après l'Agence nationale de la sécurité des systèmes d'information, ces alertes de cybersécurité n'ont eu que de "faibles impacts".

franceinfo: sport avec AFP
France Télévisions - Rédaction Sport

Publié le 10/09/2024 18:20 | Temps de lecture : 1 min

Cyberattaque à l'hôpital de Cannes : cartes d'identité, bilans médicaux, bulletins de salaire... 61 gigaoctets de données publiées sur internet

L'activité de l'hôpital est presque revenue à la normale et le système d'information est en train d'être rétabli.

franceinfo - avec France Bleu Azur
Radio France

Publié le 02/05/2024 14:55 | Temps de lecture : 1 min

Cyberattaque à l'hôpital d'Armentières : 300 000 patients concernés par le vol de données

L'établissement avait dû fermer temporairement ses urgences à la suite de cette cyberattaque le 11 février.

franceinfo avec AFP
France Télévisions

Publié le 28/02/2024 18:51 | Mis à jour le 28/02/2024 19:14 | Temps de lecture : 1 min

CYBERSÉCURITÉ \ FORMATION \ DONNÉES PERSONNELLES

Cybersécurité : l'université Paris-Saclay touchée par un ransomware

L'université francilienne a annoncé avoir été visée par une attaque par ransomware le 11 août. Un incident qui intervient une semaine après une autre cyberattaque contre plus de 40 établissements culturels en France.

Yoann Bourgin
13 août 2024 | 10h30

Cultura victime d'une cyberattaque, les données de 1,5 million de clients dérobées

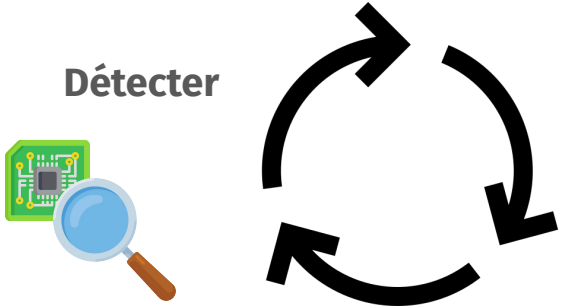
La fuite de données concerne le nom, prénom, numéro de téléphone, adresse e-mail et postale ainsi que le contenu des commandes des clients. Les mots de passe et données bancaires n'ont en revanche pas été compromis, selon l'enseigne.

franceinfo avec AFP
France Télévisions

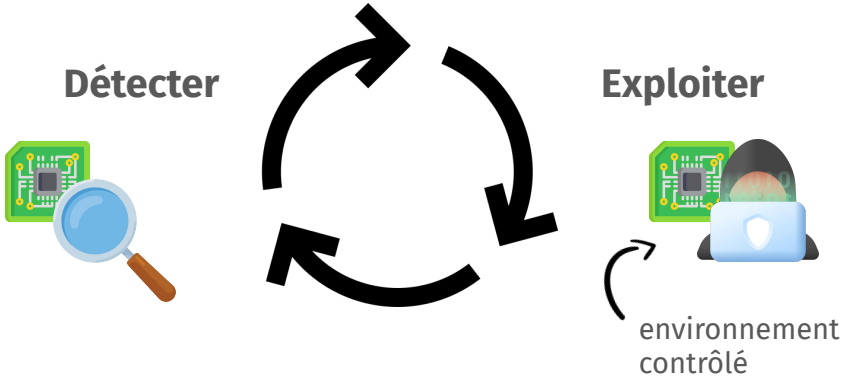
Publié le 10/09/2024 20:39 | Temps de lecture : 1 min

La sécurité n’est pas qu’une affaire de logiciels !

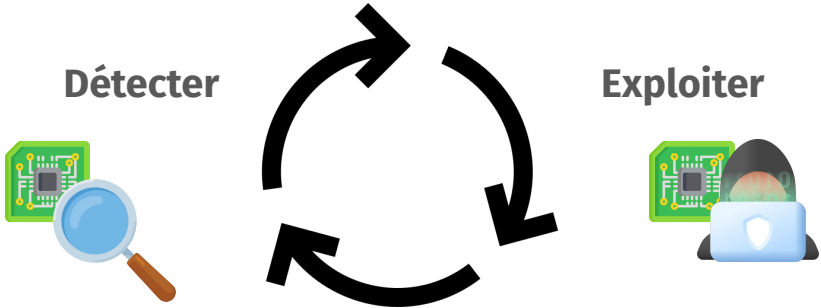
Ma recherche : sécurité à l'interface entre le logiciel et le matériel



Ma recherche : sécurité à l'interface entre le logiciel et le matériel



Ma recherche : sécurité à l'interface entre le logiciel et le matériel



Attaques matérielles

Attaques matérielles

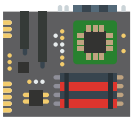
attaques actives: destruction du coffre-fort

attaques passives: écoute des mécanismes internes

Attaques matérielles

attaques actives: destruction du coffre-fort

attaques passives: écoute des mécanismes internes



attaques actives: laser, perturbation d'horloge...

attaques passives: timing, consommation de courant...

Attaques matérielles

- matériel habituellement modélisé comme couche abstraite correcte

Attaques matérielles

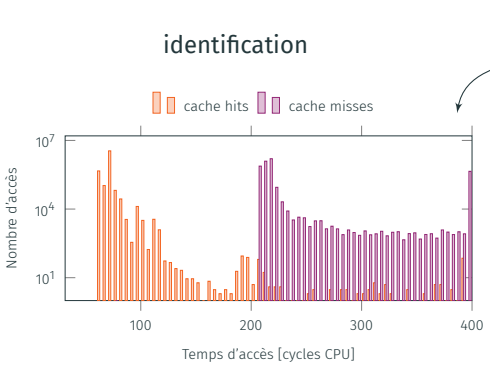
- matériel habituellement modélisé comme couche abstraite correcte, mais attaques possibles

Attaques matérielles

- matériel habituellement modélisé comme couche abstraite correcte, mais attaques possibles
  - par faute : contourner les protections logicielles par des erreurs sur le matériel
  - par canal auxiliaire : observer les effets de bord du matériel sur les calculs

Attaques matérielles

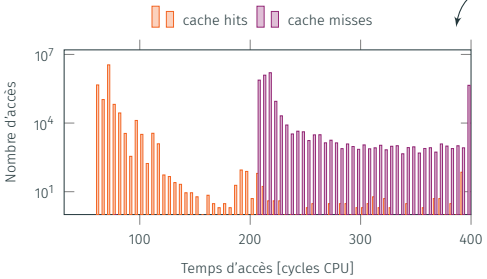
- matériel habituellement modélisé comme couche abstraite correcte, mais attaques possibles
  - par faute : contourner les protections logicielles par des erreurs sur le matériel
  - par canal auxiliaire : observer les effets de bord du matériel sur les calculs



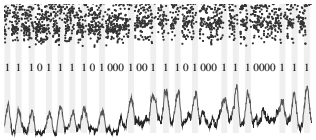
Attaques matérielles

- matériel habituellement modélisé comme couche abstraite correcte, mais attaques possibles
  - par faute : contourner les protections logicielles par des erreurs sur le matériel
  - par canal auxiliaire : observer les effets de bord du matériel sur les calculs

identification



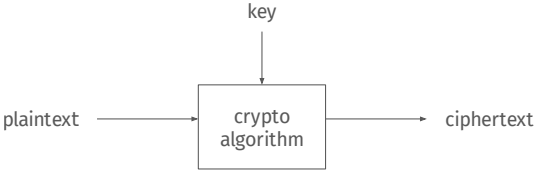
attaque



- récupérer clés secrètes, intervalles de frappes de clavier
- contourner sécurité système (ASLR)

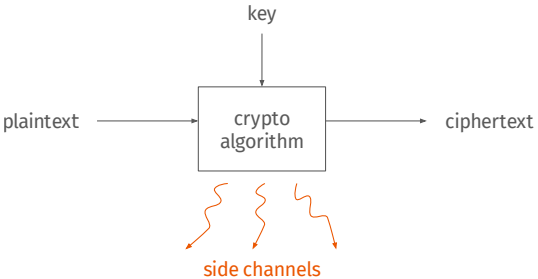
Attaques par canaux auxiliaires

- exploitent l'implémentation d'un système
- basés sur des canaux qui sont en dehors de la spécification fonctionnelle, i.e., qui ne sont pas censés véhiculer d'information utiles
- mais ces canaux peuvent faire fuiter des informations secrètes



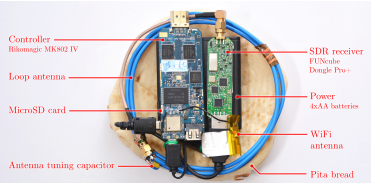
Attaques par canaux auxiliaires

- exploitent l'implémentation d'un système
- basés sur des canaux qui sont en dehors de la spécification fonctionnelle, i.e., qui ne sont pas censés véhiculer d'information utiles
- mais ces canaux peuvent faire fuiter des informations secrètes



Attaques physiques vs attaques microarchitecurales

Attaques par le matériel
a.k.a attaques physiques



Accès physique au matériel
→ appareils embarqués

VS

Attaques par le logiciel
a.k.a attaques microarchitecturales

Attaque à distance
→ systèmes complexes

De petites optimisations aux attaques par canaux auxiliaires…

- nouvelle microarchitecture tous les ans

De petites optimisations aux attaques par canaux auxiliaires…

- nouvelle microarchitecture tous les ans
- amélioration de performance $\approx 5\%$

De petites optimisations aux attaques par canaux auxiliaires…

- nouvelle microarchitecture tous les ans
- amélioration de performance $\approx 5\%$
- très petites optimisations: caches, prédicteurs de branchements…

De petites optimisations aux attaques par canaux auxiliaires…

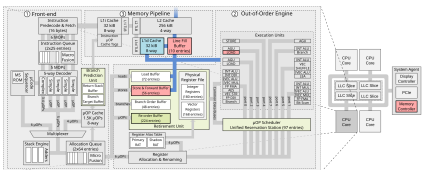
- nouvelle microarchitecture tous les ans
- amélioration de performance $\approx 5\%$
- très petites optimisations: caches, prédicteurs de branchements…
- les attaques viennent de ces optimisations

De petites optimisations aux attaques par canaux auxiliaires…

- nouvelle microarchitecture tous les ans
- amélioration de performance $\approx 5\%$
- très petites optimisations: caches, prédicteurs de branchements…
- les attaques viennent de ces optimisations
- un attaquant obtient des informations d'un processus victime (vulnérable) en espionnant l'usage du matériel

Attaques par canaux auxiliaires : les deux faces d'une même pièce

Matériel



&

Logiciel

Algorithm 1: Square-and-multiply exponentiation

Input: base b, exponent e, modulus n

Output: $b^e \mod n$

$X \leftarrow 1$

for $i \leftarrow bitlen(e)$ **downto** 0 **do**

  $X \leftarrow \text{multiply}(X, X)$

  if $e_i = 1$ **then**

    $X \leftarrow \text{multiply}(X, b)$

  end

end

return X

Questions de recherche

RQ1. Quels composants matériels sont vulnérables...

... et comment les exploiter pour faire fuir des données ?

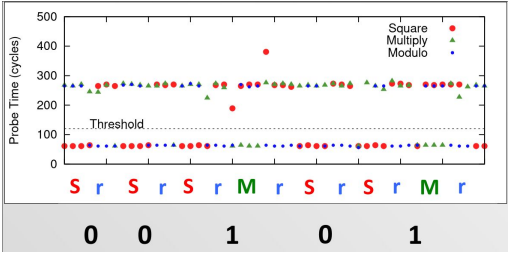
RQ2. Quelles implémentations logicielles sont vulnérables...

... et comment mener ces attaques en pratique ?

RQ1 : Un exemple d'attaque

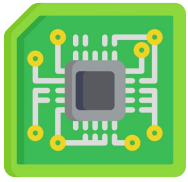
Algorithm 1: GnuPG 1.4.13 Square-and-multiply exponentiation
Input: base c, exponent d, modulus n
Output: $c^d \mod n$
$X \leftarrow 1$
for $i \leftarrow bitlen(d)$ **downto** 0 **do**
  $X \leftarrow \text{square}(X)$
  $X \leftarrow X \bmod n$
  if $d_i = 1$ **then**
    $X \leftarrow \text{multiply}(X, c)$
    $X \leftarrow X \bmod n$
  end
end
return X

valeur secrète !

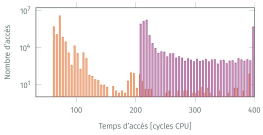


Qu’est-ce qui vient de se passer ?

attaque sur le cache

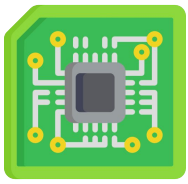


exploite les différences de temps des accès à la mémoire

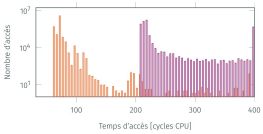


Qu’est-ce qui vient de se passer ?

attaque sur le cache

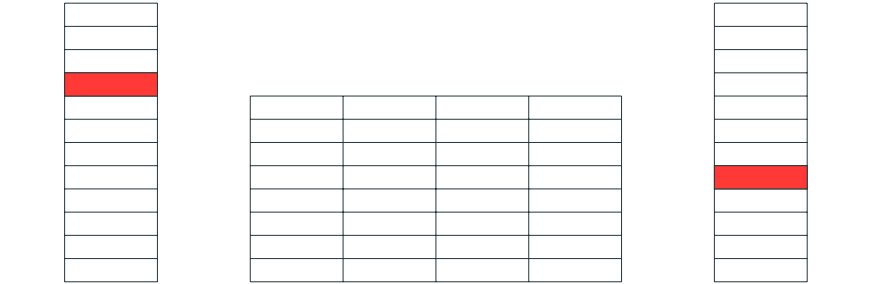


exploite les différences de temps des accès à la mémoire



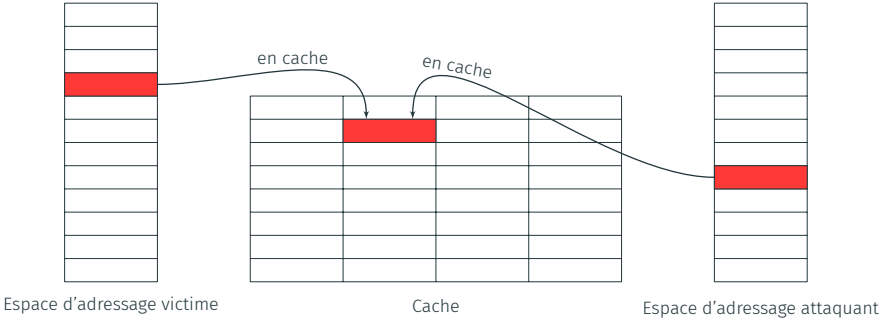
attaquant surveille les lignes accédées par la victime, pas le contenu

L’attaque sur le cache Flush+Reload



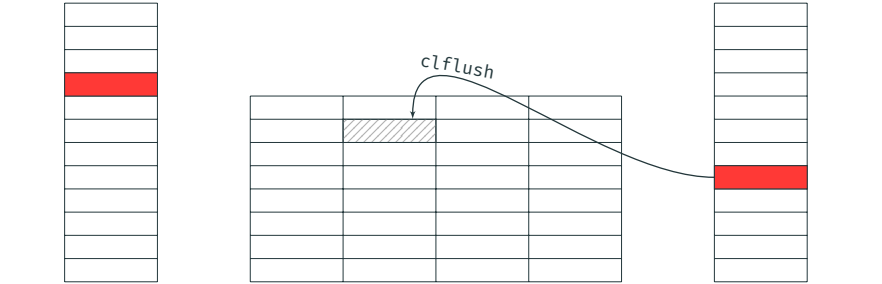
#1 : Mémoire partagée entre l’attaquant et la victime → cache partagé

L’attaque sur le cache Flush+Reload



#1 : Mémoire partagée entre l’attaquant et la victime $\rightarrow$ cache partagé

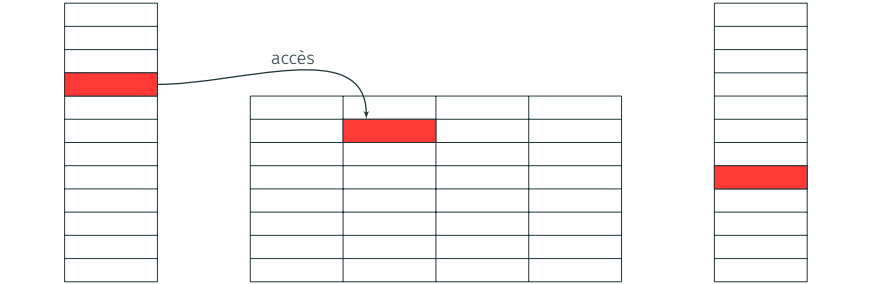
L’attaque sur le cache Flush+Reload



#1 : Mémoire partagée entre l’attaquant et la victime → cache partagé

#2 : Attaquant flushe la ligne de cache partagée

L’attaque sur le cache Flush+Reload

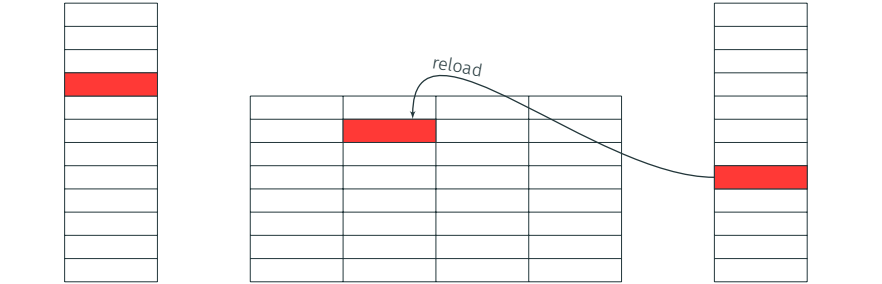


#1 : Mémoire partagée entre l’attaquant et la victime → cache partagé

#2 : Attaquant flushe la ligne de cache partagée

#3 : Victime accède à une donnée

L’attaque sur le cache Flush+Reload



#1 : Mémoire partagée entre l’attaquant et la victime → cache partagé

#2 : Attaquant flushe la ligne de cache partagée

#3 : Victime accède à une donnée

#4 : Attaquant recharge (reload) la ligne : apprend l’accès de la victime

Flush+Reload en pratique ?

```
 1  int probe(char *adrs) {
 2    volatile unsigned long time;
 3
 4    asm __volatile__ (
 5      "  mfence             \n"
 6      "  lfence             \n"
 7      "  rdtsc              \n"
 8      "  lfence             \n"
 9      "  movl %%eax, %%esi  \n"
10      "  movl (%1), %%eax   \n"
11      "  lfence             \n"
12      "  rdtsc              \n"
13      "  subl %%esi, %%eax  \n"
14      "  clflush 0(%1)      \n"
15      : "=a" (time)
16      : "c" (adrs)
17      :  "%esi", "%edx");
18    return time < threshold;
19  }
```

Flush+Reload en pratique ?

```
1  int probe(char *adrs) {
2    volatile unsigned long time;
3
4    asm __volatile__ (
5      "  mfence             \n"
6      "  lfence             \n"
7      "  rdtsc              \n"
8      "  lfence             \n"
9      "  movl %%eax, %%esi  \n"
10     "  movl (%1), %%eax   \n"
11     "  lfence             \n"
12     "  rdtsc              \n"
13     "  subl %%esi, %%eax  \n"
14     "  clflush 0(%1)      \n"
15       : "=a" (time)
16       : "c" (adrs)
17       :  "%esi", "%edx");
18   return time < threshold;
19 }
```

horloge

accès mémoire

horloge

flush cache

Conclusion RQ1

Translation look-aside buffer

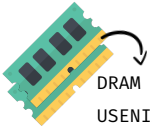
USENIX Sec'18

CPU Ports

S&P'19

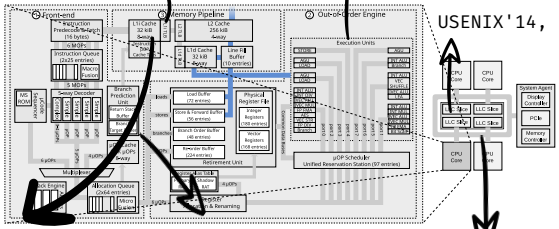
LLC attacks

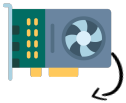
USENIX'14, S&P'15



DRAM

USENIX Sec'16





GPU

S&P'18

L1d, L1i, L2 cache

BSDCon'05, CT-RSA'06,

ASIACCS'20

Branch Prediction

CT-RSA'07

Ring Interconnect

USENIX Sec'21, DIMVA'21

État de l'art aujourd'hui: chaque composant partagé par deux processus est un potentiel vecteur d'attaque par canal auxiliaire

RQ2 : Comment on fixe ça dans les logiciels ?

Problème ?

Vulnérabilité canal auxiliaire

Tout branchement ou accès mémoire dépendant d'un secret

RQ2 : Comment on fixe ça dans les logiciels ?

💣 Problème ?

Vulnérabilité canal auxiliaire

Tout branchement ou accès mémoire dépendant d'un secret

→

💡 Solution !

Programmation *constant time*

Pas de banchement ou d'accès mémoire dépendant d'un secret

RQ2 : Comment on fixe ça dans les logiciels ?

Problème ?

Vulnérabilité canal auxiliaire

Tout branchement ou accès mémoire dépendant d'un secret

→

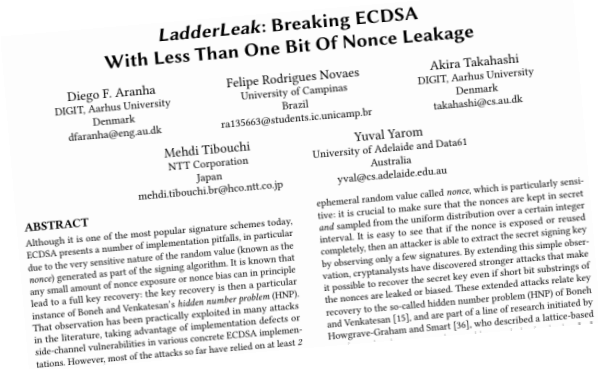
Solution !

Programmation *constant time*

Pas de banchement ou d'accès mémoire dépendant d'un secret

Ça a l'air facile non ?

Des attaques partout...



***LadderLeak*: Breaking ECDSA With Less Than One Bit Of Nonce Leakage**

Diego F. Aranha
DIGIT, Aarhus University
Denmark
dfaranha@eng.au.dk

Felipe Rodrigues Novaes
University of Campinas
Brazil
ra135663@students.ic.unicamp.br

Akira Takahashi
DIGIT, Aarhus University
Denmark
takahashi@cs.au.dk

Mehdi Tibouchi
NTT Corporation
Japan
mehdi.tibouchi.br@hco.ntt.co.jp

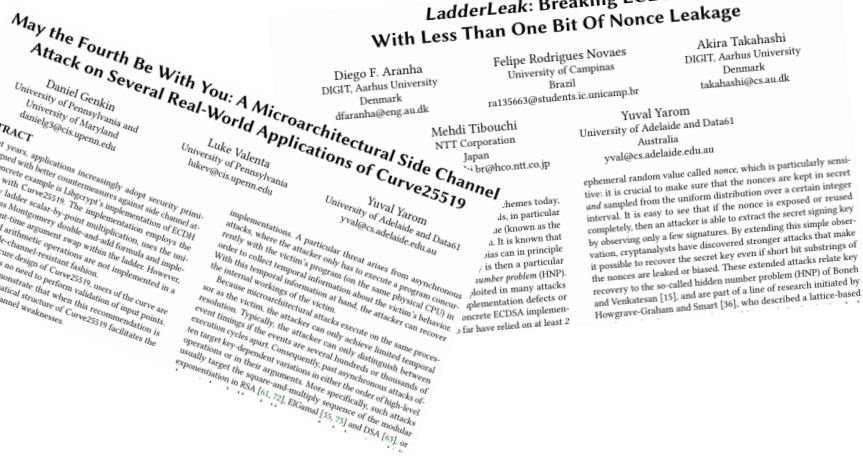
Yuval Yarom
University of Adelaide and Data61
Australia
yval@cs.adelaide.edu.au

ABSTRACT

Although it is one of the most popular signature schemes today, ECDSA presents a number of implementation pitfalls, in particular due to the very sensitive nature of the random value (known as the *nonce*) generated as part of the signing algorithm. It is known that any small amount of nonce exposure or nonce bias can in principle lead to a full key recovery: the key recovery is then a particular instance of Boneh and Venkatesan's *hidden number problem* (HNP). That observation has been practically exploited in many attacks in the literature, taking advantage of implementation defects or side-channel vulnerabilities in various concrete ECDSA implementations. However, most of the attacks so far have relied on at least 2

ephemeral random value called *nonce*, which is particularly sensitive: it is crucial to make sure that the nonces are kept in secret *and* sampled from the uniform distribution over a certain integer interval. It is easy to see that if the nonce is exposed or reused completely, then an attacker is able to extract the secret signing key by observing only a few signatures. By extending this simple observation, cryptanalysts have discovered stronger attacks that make it possible to recover the secret key even if short bit substrings of the nonces are leaked or biased. These extended attacks relate key recovery to the so-called hidden number problem (HNP) of Boneh and Venkatesan [15], and are part of a line of research initiated by Howgrave-Graham and Smart [36], who described a lattice-based

Des attaques partout...



May the Fourth Be With You: A Microarchitectural Side Channel Attack on Several Real-World Applications of Curve25519

Daniel Genkin
University of Pennsylvania and University of Maryland
danielg3@cis.upenn.edu

Luke Valenta
University of Pennsylvania
lukev@cis.upenn.edu

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

ABSTRACT

In recent years, applications increasingly adopt security primitives designed with better countermeasures against side channel attacks. A concrete example is Libgcrypt's implementation of ECDH encryption with Curve25519. The implementation employs the Montgomery ladder scalar-by-point multiplication, uses the unified branchless Montgomery double-and-add formula and implements a constant-time argument swap within the ladder. However, Libgcrypt's field arithmetic operations are not implemented in a constant-time side-channel-resistant fashion.

In this work, we demonstrate that when this recommendation is followed, the mathematical structure of Curve25519 facilitates the exploitation of side-channel weaknesses.

implementations. A particular threat arises from asynchronous attacks, where the attacker only has to execute a program concurrently with the victim's program (on the same physical CPU) in order to collect temporal information at hand, about the victim's behavior. With this temporal information at hand, the attacker can recover the internal workings of the victim.

Because microarchitectural attacks execute on the same processor as the victim, the attacker can only achieve limited temporal resolution. Typically, the attacker can only distinguish between event timings if the events are several hundreds or thousands of execution cycles apart. Consequently, past asynchronous attacks often target key-dependent variations in either the order of high-level operations or in their arguments. More specifically, such attacks usually target the square-and-multiply sequence of the modular exponentiation in RSA [61, 72], ElGamal [58, 75] and DSA [63], or

LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage

Diego F. Aranha
DIGIT, Aarhus University
Denmark
dfaranha@eng.au.dk

Felipe Rodrigues Novaes
University of Campinas
Brazil
ra135663@students.ic.unicamp.br

Akira Takahashi
DIGIT, Aarhus University
Denmark
takahashi@cs.au.dk

Mehdi Tibouchi
NTT Corporation
Japan
br@hco.ntt.co.jp

Yuval Yarom
University of Adelaide and Data61
Australia
yval@cs.adelaide.edu.au

ephemeral random value called *nonce*, which is particularly sensitive: it is crucial to make sure that the nonces are kept in secret *and* sampled from the uniform distribution over a certain integer interval. It is easy to see that if the nonce is exposed or reused completely, then an attacker is able to extract the secret signing key by observing only a few signatures. By extending this simple observation, cryptanalysts have discovered stronger attacks that make it possible to recover the secret key even if short bit substrings of the nonces are leaked or biased. These extended attacks relate key recovery to the so-called hidden number problem (HNP) of Boneh and Venkatesan [15], and are part of a line of research initiated by Howgrave-Graham and Smart [36], who described a lattice-based

Des attaques partout...

May the Fourth Be With You: A Microarchitectural Side Channel Attack on Several Real-World Applications of Curve25519

Daniel Genkin
University of Pennsylvania and
University of Maryland
danielg3@cis.upenn.edu

Luke Valenta
University of Pennsylvania
lukev@cis.upenn.edu

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

ABSTRACT

In recent years, applications increasingly adopt security primitives designed with better countermeasures against side channel attacks. A concrete example is Libgcrypt's implementation of ECDH encryption with Curve25519. The implementation employs the Montgomery ladder scalar-by-point multiplication, uses the unified, branchless Montgomery double-and-add formula and implements a constant-time argument swap within the ladder. However, Libgcrypt's field arithmetic operations are not implemented in a constant-time side-channel-resistant fashion.

Based on the secure design of Curve25519, users of the curve are advised that there is no need to perform validation of input points. In this work we demonstrate that when this recommendation is followed, the mathematical structure of Curve25519 facilitates the exploitation of side-channel weaknesses.

implementations. A particular threat arises from asynchronous attacks, where the attacker only has to execute a program concurrently with the victim's program (on the same physical CPU) in order to collect temporal information about the victim's behavior. With this temporal information at hand, the attacker can recover the internal workings of the victim.

Because microarchitectural attacks execute on ... sor as the victim, the attacker can only ... resolution. Typically, the attack... event timings if the eve... execution cycles apart... ten target key-depende... operations or in their a... usually target the square... exponentiation in RSA [61...

***LadderLeak*: Breaking ECDSA With Less Than One Bit Of Nonce Leakage**

Diego F. Aranha
DIGIT, Aarhus University
Denmark
dfaranha@eng.au.dk

Felipe Rodrigues Novaes
University of Campinas
Brazil
ra135663@students.ic.unicamp.br

Akira Takahashi
DIGIT, Aarhus University
Denmark
takahashi@cs.au.dk

Mehdi Tibouchi
NTT Corporation
Japan
...br@hco.ntt.co.jp

Yuval Yarom
University of Adelaide and Data61
Australia
yval@cs.adelaide.edu.au

...chemes today, ...ds, in particular ...ue (known as the ...n. It is known that ...bias can in principle ...y is then a particular ...number problem (HNP). ...ploited in many attacks ...plementation defects or ...oncrete ECDSA implemen... ...o far have relied on at least 2

ephemeral random value called *nonce*, which is particularly sensitive: it is crucial to make sure that the nonces are kept in secret *and* sampled from the uniform distribution over a certain integer interval. It is easy to see that if the nonce is exposed or reused completely, then an attacker is able to extract the secret signing key by observing only a few signatures. By extending this simple observation, cryptanalysts have discovered stronger attacks that make it possible to recover the secret key even if short bit substrings of the nonces are leaked or biased. These extended attacks relate key recovery to the so-called hidden number problem (HNP) of Boneh and Venkatesan [15], and are part of a line of research initiated by Howgrave-Graham and Smart [36], who described a lattice-based

PARASITE: PAssword Recovery Attack against Srp Implementations in ThE wild

Daniel De Almeida Braga
daniel.de-almeida-braga@irisa.fr
Univ Rennes, CNRS, IRISA
Rennes, France

Pierre-Alain Fouque
pa.fouque@gmail.com
Univ Rennes, CNRS, IRISA
Rennes, France

Mohamed Sabt
mohamed.sabt@irisa.fr
Univ Rennes, CNRS, IRISA
Rennes, France

ABSTRACT

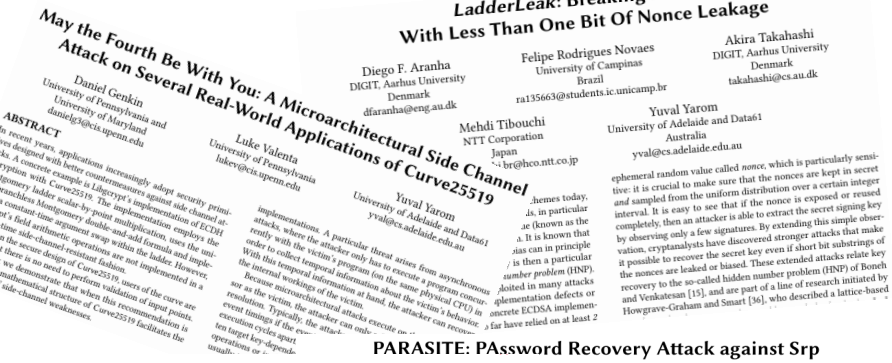
Protocols for password-based authenticated key exchange (PAKE) allow two users sharing only a short, low-entropy password to establish a secure session with a cryptographically strong key. The challenge in designing such protocols is that they must resist offline dictionary attacks in which an attacker exhaustively enumerates

KEYWORDS

SRP; PAKE; Flush+Reload; PDA; OpenSSL; micro-architectural attack

ACM Reference Format:
Daniel De Almeida Braga, Pierre-Alain Fouque, and Mohamed Sabt. 2021.

Des attaques partout...



May the Fourth Be With You: A Microarchitectural Side Channel Attack on Several Real-World Applications of Curve25519

Daniel Genkin
University of Pennsylvania and University of Maryland
danielg3@cis.upenn.edu

Luke Valenta
University of Pennsylvania
lukev@cis.upenn.edu

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

ABSTRACT

In recent years, applications increasingly adopt security primitives designed with better countermeasures against side channel attacks. A concrete example is Libgcrypt's implementation of ECDH encryption with Curve25519. The implementation employs the Montgomery ladder scalar-by-point multiplication, uses the unified Montgomery ladder formulas and implements a constant-time argument swap within the ladder. However, Libgcrypt's field arithmetic operations are not implemented in a constant-time side-channel-resistant fashion.

In this work we demonstrate that when this recommendation is followed, the mathematical structure of Curve25519 facilitates the exploitation of side-channel weaknesses.

LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage

Diego F. Aranha
DIGIT, Aarhus University
Denmark
dfaranha@eng.au.dk

Felipe Rodrigues Novaes
University of Campinas
Brazil
ra135663@students.ic.unicamp.br

Akira Takahashi
DIGIT, Aarhus University
Denmark
takahashi@cs.au.dk

Mehdi Tibouchi
NTT Corporation
Japan
br@hco.ntt.co.jp

Yuval Yarom
University of Adelaide and Data61
Australia
yval@cs.adelaide.edu.au

ephemeral random value called *nonce*, which is particularly sensitive: it is crucial to make sure that the nonces are kept in secret *and* sampled from the uniform distribution over a certain integer interval. It is easy to see that if the nonce is exposed or reused completely, then an attacker is able to extract the secret signing key by observing only a few signatures. By extending this simple observation, cryptanalysts have discovered stronger attacks that make it possible to recover the secret key even if short bit substrings of the nonces are leaked or biased. These extended attacks relate key recovery to the so-called hidden number problem (HNP) of Boneh and Venkatesan [15], and are part of a line of research initiated by Howgrave-Graham and Smart [36], who described a lattice-based

Side-Channel Analysis of SM2: A Late-Stage Featurization Case Study

Nicola Tuveri
Tampere University of Technology
Tampere, Finland
nicola.tuveri@tut.fi

Sohaib ul Hassan
Tampere University of Technology
Tampere, Finland
sohaibulhassan@tut.fi

Cesar Pereida García
Tampere University of Technology
Tampere, Finland

Billy Bob Brumley
Tampere University of Technology
Tampere, Finland

PARASITE: PAssword Recovery Attack against Srp ntations in ThE wild

Pierre-Alain Fouque
pa.fouque@gmail.com
Univ Rennes, CNRS, IRISA
Rennes, France

Mohamed Sabt
mohamed.sabt@irisa.fr
Univ Rennes, CNRS, IRISA
Rennes, France

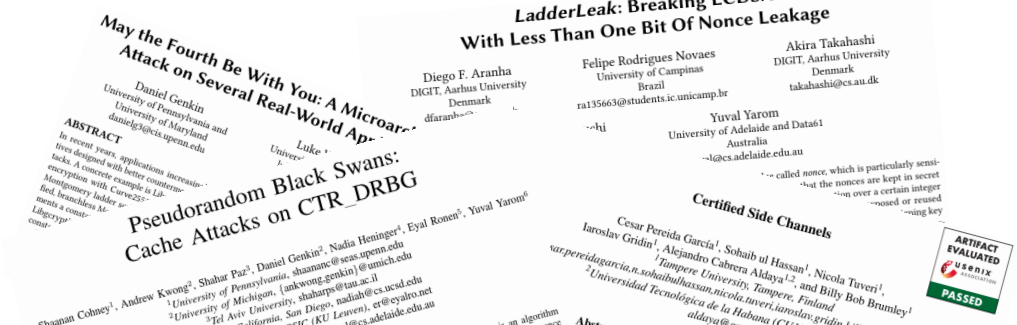
KEYWORDS

SRP; PAKE; Flush+Reload; PDA; OpenSSL; micro-architectural attack

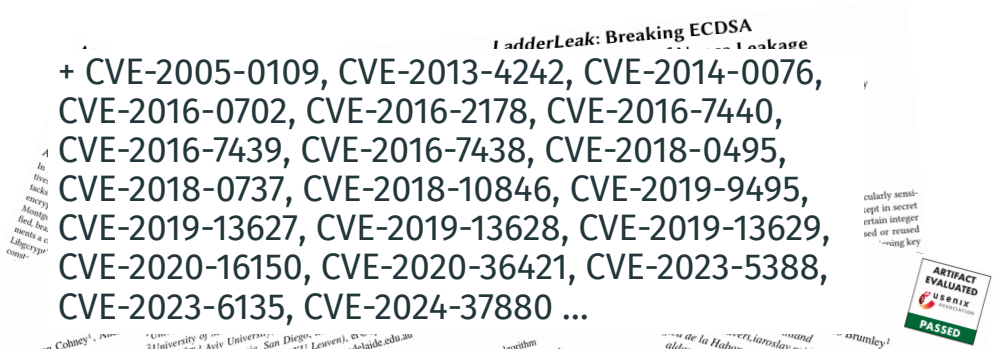
ACM Reference Format:
Daniel De Almeida Braga, Pierre-Alain Fouque, and Mohamed Sabt. 2021.

Des attaques partout...

Des attaques partout...

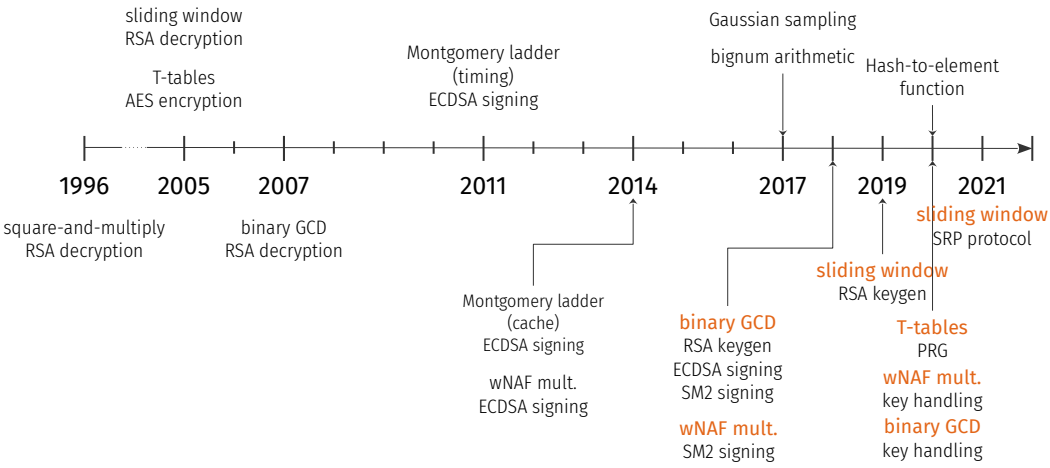


Des attaques partout...



Des attaques partout...

Comparaison de vulnérabilités récentes (2017-2022) et anciennes



Les mêmes vulnérabilités reviennent à la surface, pourquoi ?

Nouveaux contextes

- génération de clés
- traitement des clés
- génération de nombres aléatoires

- le code vulnérable reste dans la bibliothèque (surtout pour OpenSSL)

Les mêmes vulnérabilités reviennent à la surface, pourquoi ?

Nouveaux contextes

- génération de clés
- traitement des clés
- génération de nombres aléatoires

Nouvelles bibliothèques

- implémentation RSA ”sliding window” dans MbedTLS (2017)
- attaques Bleichenbacher dans MbedTLS, s2n, et NSS (2019)

- le code vulnérable reste dans la bibliothèque (surtout pour OpenSSL)
- vulnérabilités trouvées dans OpenSSL mais patchs non propagés aux autres bibliothèques

Conclusion RQ2

La plupart des vulnérabilités proviennent de code déjà identifié comme étant vulnérable

Beaucoup d'outils (académiques) existants, mais

- peu connus, peu utilisés, peu utilisables
- toujours des limitations techniques
- des architectures de bibliothèques cryptographiques très complexes

Conclusions

- papier séminal par Kocher en 1996 : près de 30 ans de recherche
- domaine en expension : forte augmentation du nombre d'articles depuis 2015
- les attaques microarchitecturales nécessitent :
  - la compréhension et le contrôle bas niveau du matériel → microarchitecture, rétro-ingénierie
  - la compréhension très fine du logiciel → analyse de programme, compilation, cryptographie

→ il y a du travail dans toutes les couches d'abstraction !

Merci !

Contact

- clementine.maurice@cnrs.fr
- @BloodyTangerine / @bloody-tangerine.bsky.social

Cybersécurité à la loupe : du logiciel au matériel

Clémentine Maurice, Chargée de recherche CNRS
Laboratoire CRIStAL, équipe Spirals

1er Octobre 2025—Bar des sciences, Université de Lille