Les questions scientifiques de la sécurité informatique

Gérard Berry

Professeur au Collège de France
Chaire Algorithmes, machines et langages
Académie des sciences, Académie des technologies

http://www.college-de-france.fr/site/gerard-berry

Cours 4, 13/02/2019

COLLÈGE DE FRANCE
1530

2019, World Economic Forum

Classement des dangers économiques majeurs :

1. Evénements climatiques extrêmes
2. Désastres naturels
3. Attaques informatiques contre les infrastructures
4. Vols de données et de comptes
5. …

Note : sujet déjà abordé au Collège de France

- Séminaire de Jacques Stern dans mon cours, 28/03/2008
- Cours de Martin Abadi, chaire Informatique et sciences numériques 2010-2011
  https://www.college-de-france.fr/site/martin-abadi/course-2010-2011.htm
- Séminaire de Dominique Bolignano dans mon cours, 11/03/2015
  https://www.college-de-france.fr/site/gerard-berry/seminar-2015-03-11-17h30.htm
- Séminaire de Véronique Cortier dans mon cours, 25/03/2015
  https://www.college-de-france.fr/site/gerard-berry/seminar-2015-03-25-17h30.htm
- Séminaire de Stéphanie Delaune dans mon cours, 06/04/2016
  https://www.college-de-france.fr/site/gerard-berry/seminar-2016-04-06-17h30.htm

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Le point de vue de l'utilisateur moyen

1. Je veux utiliser mes appareils (ordinateurs, tablettes, téléphones) sans que personne ne puisse y entrer sans mon consentement, sauf pour ceux que je désigne (comme pour ma maison)

2. Quand j'ai un compte sur un site Internet, je veux que mon compte et mes moyens de paiement soient protégés. (comme pour mon portefeuille quand il est dans ma poche).

3. On m'a dit qu'il y avait des dangers, et de ne pas répondre aux mails ni cliquer partout, mais je ne sais pas trop où me renseigner et je ne veux pas me casser la tête….

Le point de vue de l'utilisateur plus averti

Après beaucoup de sueur et de consultation de sites web variés, j'ai compris d'autres dangers :

- Le vol de mes données de comptes sur des sites pourris
- Le cassage de mes mots de passe, que je croyais bons. Et si je ne mets jamais le même, comment m'en souvenir ?
- Le pistage permanent de ma localisation par des tas de gens que je ne sais même pas détecter
- Le pistage de tout ce que je fais sur mon navigateur ou sur mon ordi, par les mêmes ou d'autres
- L'interception ou le trucage de mes communications
- Et même carrément le vol de mon indentité !

Ça augmente tous les jours !

- Les faux messages, qui ont pourtant l'air parfaitement vrais, et sont de l'hameçonnage. Au fait, comment fait-on pour signer ?
- En plus des virus, les programmes intrus qui utilisent ma propre puissance de calcul pour le profit d'autres
- La prise de contrôle de mes objets connectés, dont on m'avait pourtant dit qu'il étaient parfaitement sûrs
- Mon garagiste m'a même fait acheter une cage de Faraday pour mettre la carte-clef de ma belle voiture, car on peut la sniffer de l'extérieur et partir avec la voiture !

La liste a l'air de ne jamais s'arrêter…

Je fais quoi, en face de tout ça ?

La point de vue du DSI d'avant (~fictif)

Les bonnes pratiques de base sont suffisantes :

- avoir des PCs et OS éprouvés et de bons antivirus
- ne pas faire les mises à jour trop vite, car certains programmes ne vont plus marcher et ça va me retomber dessus. Les vieux systèmes sont les meilleurs !
- faire de bonnes sauvegardes sur bandes, et les garder dans un coffre-fort tolérant l'eau et le feu
- avoir un bon outillage pour la gestion des comptes et des utilisateurs, de bonnes procédures pour l'interation avec eux, et des ingénieurs compétents
- pour le reste, me renseigner chez les copains et faire comme eux

Se limiter à ça est un pur bonheur pour les attaquants modernes ! par exemple, garder Windows XP…

Le point de vue d'un DSI récent (~fictif)

- Un matin : *Damned !* Tous nos ordinateurs sous XP se sont fait crypter cette nuit par un virus, apparemment Wannacry !
- Un peu plus tard : *By Jove !* Nos machines-outils aussi étaient sous XP, elles sont toutes bloquées !
- Le soir : *Good Lord !* Nos sauvegardes ne marchent pas ! On ne les avait jamais testées à l'échelle réelle…

Une semaine après, le point de vue du PDG :
The Devil ! ça nous a coûté 200 millions d'Euros…
Il faut changer tout ça, et vite !

Juillet 2017 :
La police de Londres passe ses 18 000 ordinateurs de Windows XP à Windows Vista....

Le point de vue des attaquants

Quelle époque bénie !

- Tout est connecté, et il y a des trous partout : mots de passes triviaux, failles d'OS, de navigateurs, de circuits, etc.
- Tout est attaquable, particuliers, états et entreprises
- Même pas besoin de les chercher soi-même, on trouve plein d'attaques sur le web pour trois francs six sous, il suffit de savoir les assembler et de les lancer par robots logiciels
- Comme les congrès de sécurité sont publics, on sait ce que trouvent les chercheurs
- Rien n'est plus facile que de trouver des sponsors !
- Le faire pour son pays ou pour l'argent, ce n'est même plus incompatible
- …

Le point de vues des autorités

Séminaire à suivre de Guillaume Poupard,

Directeur général de l'ANSSI

(Agence nationale de sécurité des systèmes informatiques)

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Quelques types d'attaques

Vol ou modification de données
- identité, mots de passe, moyens de paiements, dossier médical, fichiers et mails personnels ou d'entreprises, …

Hameçonnage

Pistage, détection analyse systématique du comportement

Vol de puissance de calcul
- si votre ordi est lent, n'est-il pas en train de miner du Bitcoin ?

Interception ou piratage de communications
- transactions sur Internet, communications vocales ou vidéo, espionnage, …

Attaques sur les systèmes
- déni de service
- destruction ou cryptage des fichiers (particuliers ou entreprises)
- attaques sur hôpitaux, avions, trains, voitures, contrôle aérien réseaux électriques, feux rouges des villes, objets connectés, …

Vols de données mal protégées

- 2019 :
  - 773 M* de paires adresse / mot de passe
    trouvées sur un site de hackers – dont 6 pour moi !
  - Singapour : 14 000 données médicales de malades du SIDA
- 2018
  - Mariott-Starwood : 500 M réservations de 327 M clients
- 2017
  - Equifax (certification de crédits US) : 145,5 M comptes
- …

Sans compter les cas non rendus publics !

* M = million

Le passage à la grande échelle

2007 : attaque de déni de service de grande envergure sur l'Estonie, importante paralysie du pays

2010 : fameux virus Stuxnet, centrale nucléaire de Bouchehr

2014 : *Sony Pictures Entertainment*, vol massif de films et de données d'employés

2016 : rançongiciel Petya sur Windows, expansion rapide

2016 : attaque massive du DNS Dyn, par recrutement de millions d'objets connectés mal protégés (malware Mirai)

2017 : Trois attaques mondiales successives par rancongiciels sur Windows Wannacry, Adilkuzz, NotPetya paralysent hôpitaux, grandes entreprises, banques et des administrations

2018 : piratage massif de comptes bancaires chez HSBC

2019 : publication de milliers de documents confidentiels de politiciens allemands

Failles trouvées par les chercheurs

2019 :

- Amadeus (France, 140 compagnies aériennes) : voir et changer les dossiers
- Apple Face Time multiparticipants transmet la vidéo avant l'acceptation de la connection

2018 :

- US Postal a exposé 60 M comptes pendant un an
- Des chercheurs ont attaqué Alexa d'Amazon pour épier les conversations
- Recommandation d'utiliser des cages de Faraday pour y metttre ses cartes-clefs de voitures haut de gamme !

Déjà mentionnés :

- prises de contrôle des Jeep Cherokee, en domotique des Samsung Smart Things, etc.

Saurons nous vraiment sécuriser l'Internet des objets ?

Ex. : failles majeures détectées dans les OS

- 2017 : 3 vulnérabilités majeures sur les machines virtuelles de VMware
- 2018 : Linux se crashe si on envoie un message vide au processus systemd (nº 1)
- 2018 : IoS se crashe quand il reçoit un SMS réduit à un certain caractère spécial
- 2018 : Meltdown et Spectre, attaques majeures de tous les OS d'ordinateurs et téléphones sur Pentium, AMD ou ARM haut de gamme (voir plus loin)

Et des vulnérabilités importantes corrigées à chaque nouvelle version de Windows, MacosX, iOS, Linux et Android

Raison nº 1 : efficacité → non validation des accès mémoire

Il est indispensable de faire immédiatement les mises à jour

Halte aux légendes : tous les OS sont concernés

Infrastructure d'accessibilité
macOS High Sierra 10.13.5, mise à jour de sécurité 2018-003 pour Sierra

Conséquence : une application malveillante
peut exécuter un code arbitraire avec des privilèges système.

Description : l'infrastructure d'accessibilité présentait un
problème de divulgation d'informations.

Ce problème a été résolu par une meilleure gestion de la mémoire.
CVE-2018-4196 : G. Geshev en collaboration avec le programme
Zero Day Initiative de Trend Micro, chercheur anonyme

FontParser
…macOS Sierra 10.12.6, macOS High Sierra 10.13.4

Conséquence : le traitement d'un fichier de polices malveillant peut
entraîner l'exécution arbitraire de code.

Description : un problème de corruption de la mémoire a été résolu par une meilleure validation.
CVE-2018-4211 : Proteas de la Qihoo 360 Nirvan Team

iBooks
Disponible pour : macOS High Sierra 10.13.4

Conséquence : un attaquant bénéficiant d'une position privilégiée sur le réseau peut être en mesure de détourner des invites de mot de passe

Description : un problème de validation des entrées a été résolu par une meilleure validation des entrées.

CVE-2018-4202 : Jerry Decime

IOGraphics
Disponible pour : macOS High Sierra 10.13.4

Conséquence : une application peut exécuter un code arbitraire avec des privilèges liés au noyau.

Description : un problème de corruption de la mémoire a été résolu par une meilleure gestion de cette dernière.

CVE-2018-4236 : Zhao Qixun (@S0rryMybad) de la Qihoo 360 Vulcan Team

Mail

Disponible pour : macOS High Sierra 10.13.4

Conséquence : un attaquant peut exfiltrer le contenu d'un e-mail chiffré avec un algorithme S/MIME.

Description : un problème existait au niveau de la gestion des e-mails chiffrés. Ce problème a été résolu par une meilleure isolation de l'algorithme MIME dans les e-mails.

CVE-2018-4227 : Damian Poddebniak, de l'Université des sciences appliquées de Münster[1], Christian Dresen idem, Jens Müller de l'Université de la Ruhr à Bochum[2], Fabian Ising[1], Sebastian Schinzel[1], Jörg Schwenk[2]

…. et encore 26 autres corrections dans cette version !

La plupart des failles viennent de bugs logiciels classiques mais pas détectés par les tests fonctionnels : débordement de tableaux, corruption de mémoire, etc.

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Quatre grands types d'action

Etudier les attaques existantes

Trouver les failles avant les acteurs malveillants
- dans les chiffrements
- dans les protocoles de sécurité (ex. TLS de https sur Internet)
- dans les OS et les logiciels d'application
- et même dans les circuits électroniques

Construire de meilleurs environnements de sécurité
- exemples : réseaux de communication (5G), protocoles Internet, systèmes de votes « électroniques » prouvés

De plus en plus d'excellents chercheurs, en réseau mondial !

Loi de base : quand on trouve une attaque, donner du temps aux insustriels pour la fixer avant toute publication échoue parfois, par lassitude (FaceTime) ou fuites (Freak)

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Le chiffrement, la brique de base

Fondamental pour les mots de passe, la sécurisation des fichiers, celle des communications, etc.

1. Chiffrement symétrique :

   une seule clef k à b bits pour coder et décoder un message m

   – idéal : une bijection aléatoire $2^b \rightarrow 2^b$ – mais impraticable

   – méthodes itératives pseudo-aléatoires efficaces, p.ex. AES :

     k : 128 bits (secret), 192 ou 256 bits (top secret)

   – méthodes fondées sur le logarithme discret, plus lentes

     étant donné un nombre premier p fixé (512-1024 bits), et g un générateur ($\{g^x \bmod p\} = [\,0..p-1]$), coder n par $\{n\}_g = g^n \bmod p$

pb du log discret : trouver n connaissant $\{n\}_g$ est supposé dur (mais montré pas toujours dur)

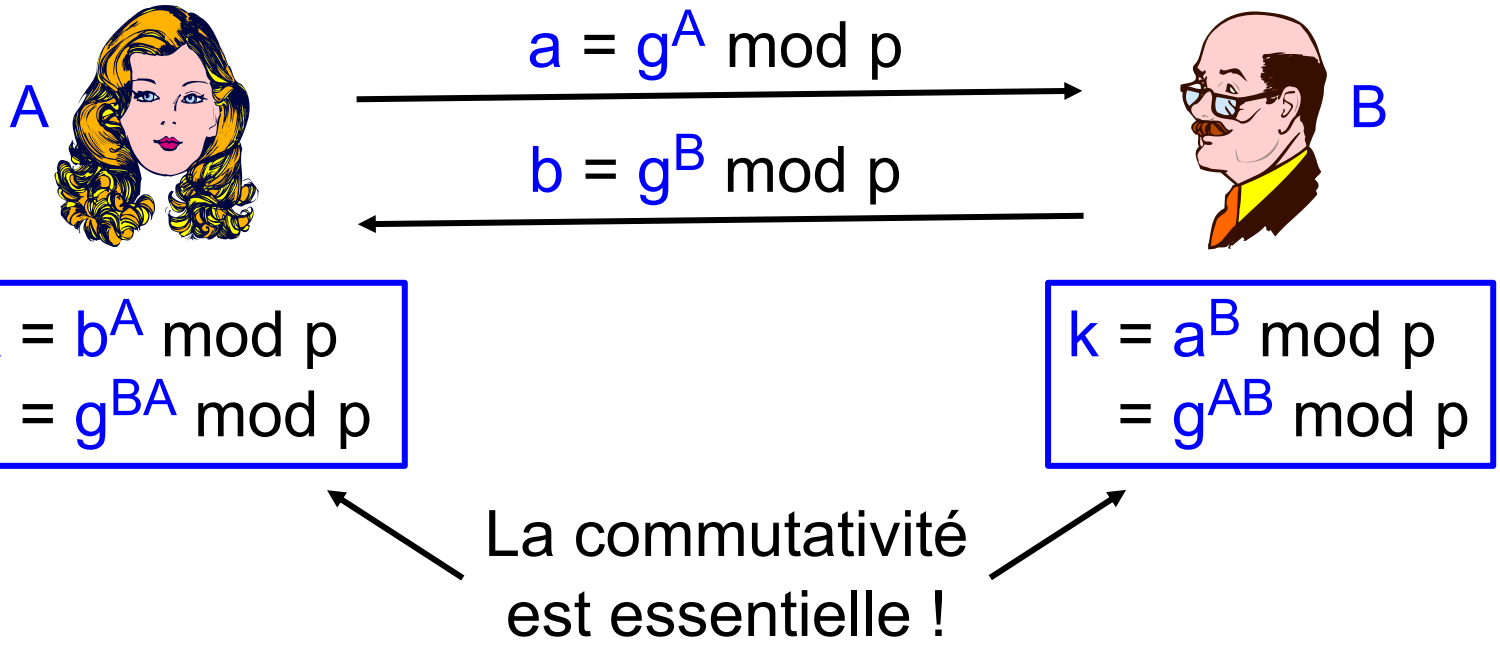
autres choix : courbes elliptiques

Echanger une clef de façon symétrique

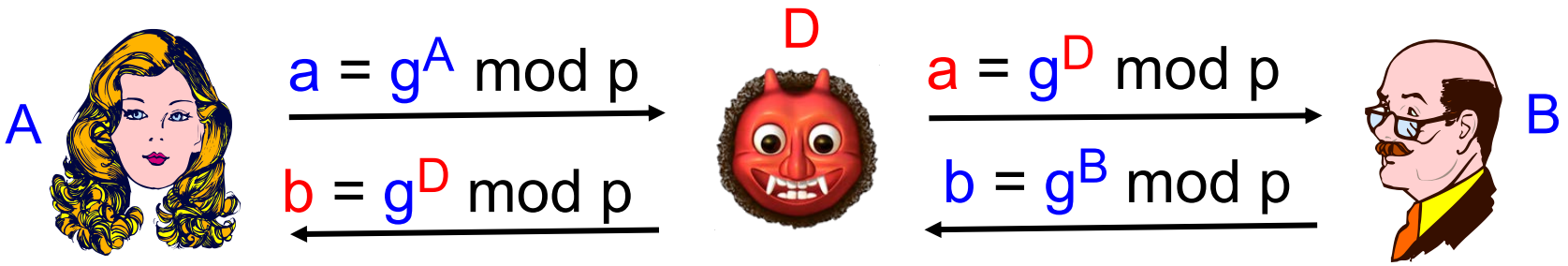
- Mais comment échanger la clef k entre Alice et Bob ?

Diffie-Hellman, 1976 :

- p fixé et public, on choisit un générateur g, public aussi
- Alice choisit une clef secrète $A \in [0..p-1]$, Bob choisit B.
- il suffit d'un échange de messages indépendants



Mais attaque « man in the middle »



kA = b^A mod p
= g^DA mod p

kA = a^D mod p
= g^AD mod p
kB = b^D mod p
= g^BD mod p

kB = a^B mod p
= g^DB mod p

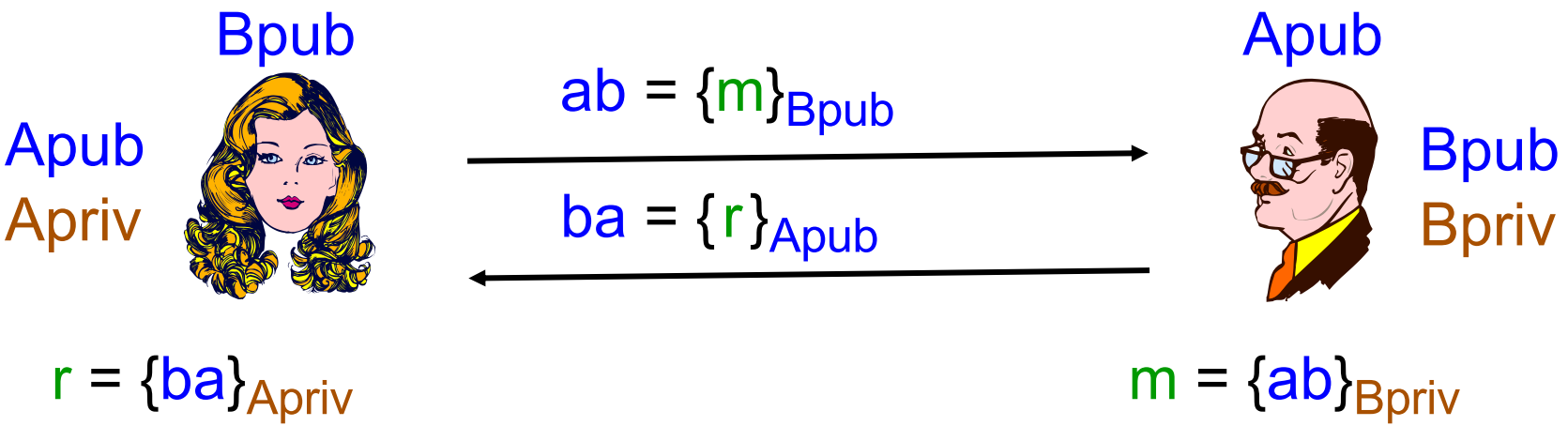
le démon sait tout !

Les attaques *man in the middle* sont nombreuses et souvent le prélude à des attaques plus fortes

Désolé, la planche de la vidéo était trop compliqée….

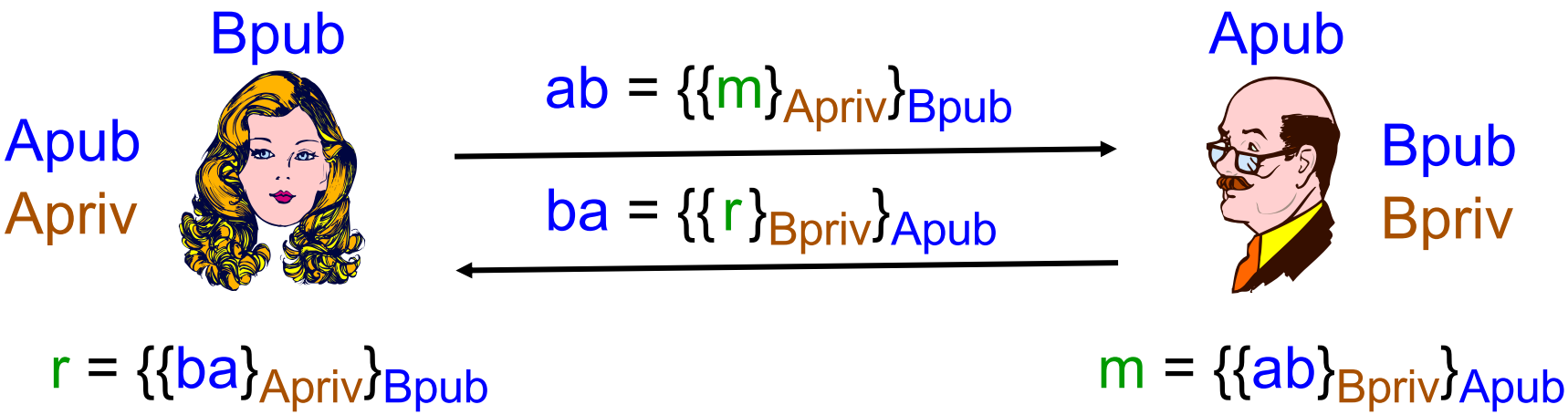
Chiffrement asymétrique : 4 cadenas !

- Idée géniale de Diffie et Hellman (et autres), implémentée aussi dans RSA (Rivest-Shamir-Adleman)
- Utiliser deux clefs, l'une Xpub publique et l'autre Xpriv privée, telles que $\{\{m\}_{Xpriv}\}_{Xpub} = \{\{m\}_{Xpub}\}_{Xpriv} = m$



Chiffrement asymétrique : 4 cadenas !

- Idée géniale de Diffie et Hellman (et autres), implémentée aussi dans RSA (Rivest-Shamir-Adleman)
- Utiliser deux clefs, l'une Xpub publique et l'autre Xpriv privée,
- telles que $\{\{m\}_{Xpriv}\}_{Xpub} = \{\{m\}_{Xpub}\}_{Xpriv} = m$
- Mais comment être sûr que le message vient d'Alice (ou Bob) ?



Trop cher en pratique, transmettre $\{m, \{Hash(m)\}_{Apriv}\}_{Bpub}$

Autres signature

Je suis l'auteur de ce message
et je ne pourrai pas dire le contraire (non répudiation)

1. Logarithme discret + hash :

El Gamal → DSA, *Digital Signature Authentification*

2. Zero-knowledge proof : prouver à son interlocuteur qu'on possède un secret, mais sans rien révéler de ce secret.

Exemples : graphe 3-coloré, chemin hamiltonien, etc.

signature : graphe 3-coloré, facile à construire, mais difficile à vérifier ! Transmettre en cachant les couleurs.

receveur : montrer les couleurs de 2 sommets reliés; si égales, NOK, sinon, proba OK ↗
recommencer n fois avec couleurs changées, proba→1
Ne donne aucune info sur la coloration !

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Les protocoles de sécurité

Comment stocker mots de passe, chiffrer les fichiers, organiser les communications sur le web et avec objets connectés, construire un bon système de vote, etc.

recherche : casser l'existant, proposer mieux, et surtout spécifier mieux et prouver avec les ressources de la logique mathématique

exemples :

faille trouvée dans le passeport électronique français (Nancy)

nombreuses failles dans TLS (*Transport Layer Security* de https) et dans les protocoles réseaux en général (une trouvée récemment dans la 5G) – le sujet est dur !

Nouveau système Bélénios pour vote vérifiable (Nancy)

Outil majeur : Provérif, vérifieur formel de Bruno Blanchet (ENS)

….

Attaques sur TLS (https)

TLS (*Transport Layer Security*) est un protocole complexe d'établissement de communications et de négotiation du chiffrement. Un problème dur : la compatibilité avec le passé ou les chiffements export grade (NSA)

- Freak (Barghavan, Fournet *et. al., 2015*), attaquesur des implémentations de TLS : touchait > 10% des sites

Dans la négotiation du chiffrement à employer, et par une attaque *man in the middle*, forcer la communication à utiliser RSA « export variant » 512 bits, qu'on sait casser

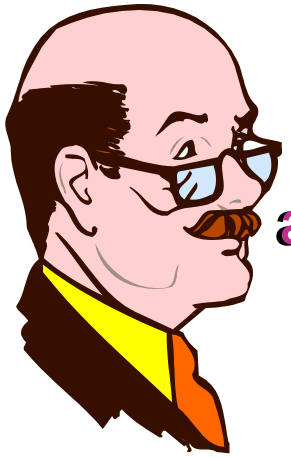
- Logjam (Nancy *et. al.*, 2015), sur la spécification de TLS : touchait 8,4% des sites

Par *man in the middle*, forcer Diffie-Hellman 512 bits avec un premier p et un générateur g bien connus, pour lequel beaucoup de modulos sont précalulables efficacement

Vers des TLS / https vérifiés ?

- MiTLS : implémentation de référence + analyse des attaques, par K. Barghavan (INRIA), C. Fournet (Microsoft Research Cambridge) *et. al.*
  https://www.mitls.org/
- Everest : développement open-source de TLS et https à spécification et code vérifiés mathématiquement (sauf pour les canaux cachés), par les mêmes + CMU (Carnegie Mellon Univ.)
  https://project-everest.github.io/

Difficulté : le wb est dynamique



Bob@cdf.fr

Annuaire

adr<Alice@cdf.fr

rep<alice@cdf.fr>

alice@cdf.fr<Bonjour, ici Bob@cdf.fr

Modélisation et vérification (partielle)

le π-calcul de Robin Milner

Alice@cdf.fr

Bob@cdf.fr<vous habitez chez vos parents

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Qu'est-ce qu'un système de vote transparent ?

1. L'urne doit être publique, et le vote vérifiable
2. Seuls les inscrits votent, et ils sont identifiés à ce moment
3. Chacun peut vérifier que les votes viennent d'inscrits, et que chacun n'a voté qu'une fois (pas de bourrage)
4. Tous doivent pouvoir vérifier le comptage final
5. Secret : l'urne ne doit pas permettre de savoir qui a voté quoi
6. Chaque votant doit pouvoir vérifier que son vote est dans l'urne et pas modifié
7. Mais il ne doit pas pouvoir prouver à son vote à un tiers (le vendre)
8. Et tout cela même si les autorités sont malhonnêtes

Problème : on ne sait pas faire tout ça en même temps…

Les pratiques actuelles

Votes nationaux :

- papier bien si le vote est contradictoire, avec autorités fiables et assesseurs attentifs ; mais pas de vérifiabilité individuelle…
- par correspondance : piètres garanties
- électronique : des pays le font, d'autres l'interdisent (NL, D)

Machines à voter aux USA : toutes montrées faciles à truquer...

- voir la page de J. Alex Halderman, https://jhalderm.com/

Votes internes du CNRS : bourrage des urnes facile (LORIA)

Votes électroniques en France : pas de transparence, voire bugs

- ex. élections ratées à l'Académie des technologies…

Le vote électronique n'est actuellement pas assez sûr pour des élections politiques nationales. Le sera-t-il un jour ? Pas sûr !

Le secret du code des machines à voter ne garantit pas leur sécurité, bien au contraire ! La plupart ne permettent pas de vérification externe incontestable, et ne résistent pas aux attaques, quelquefois même grossières…

NB: Les planches qui suivent viennent d'un exposé de Véronique Cordier, LORIA Nancy

Protocole de vote Belenios

- Développé au Loria, équipe Cassis et Caramel
  Développeur : Stéphane Glondu
- Variante de Helios, développé et utilisé à l'université de Louvain (start-up BlueKrypt)

`http://belenios.gforge.inria.fr/`

- Assure la confidentialité des votes
- Permet la transparence du scrutin
  → L'urne est publique à tout moment.
  → Les calculs (comptage, ...) sont vérifiables par tous.

Fonctionnement de Belenios (simplifié)

Phase 1 : vote

Urne		
Alice	$\{v_A\}_{\text{pub}(E)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{\text{pub}(E)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{\text{pub}(E)}$	$v_C = 0$ or 1

pub(E) : clef publique, la clef privée est partagée entre les autorités.

Fonctionnement de Belenios (simplifié)

Phase 1 : vote

$\{v_D\}_{\text{pub}(E)}$

Urne

Alice	$\{v_A\}_{\text{pub}(E)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{\text{pub}(E)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{\text{pub}(E)}$	$v_C = 0$ or 1

pub(E) : clef publique, la clef privée est partagée entre les autorités.

Fonctionnement de Belenios (simplifié)

Phase 1 : vote

Urne		
Alice	$\{v_A\}_{\text{pub}(E)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{\text{pub}(E)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{\text{pub}(E)}$	$v_C = 0$ or 1
David	$\{v_D\}_{\text{pub}(E)}$	$v_D = 0$ or 1

pub(E) : clef publique, la clef privée est partagée entre les autorités.

Fonctionnement de Belenios (simplifié)

Phase 1 : vote

Urne		
Alice	$\{v_A\}_{pub(E)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{pub(E)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{pub(E)}$	$v_C = 0$ or 1
David	$\{v_D\}_{pub(E)}$	$v_D = 0$ or 1
...	...	

Phase 2 : Dépouillement - chiffrement homomorphique (El Gamal)

$$\{v_1\}_{pub(E)} \times \cdots \times \{v_n\}_{pub(E)} = \{v_1 + \cdots + v_n\}_{pub(E)} \quad \text{car } g^a \times g^b = g^{a+b}$$

$\rightarrow$ Seul le résultat final doit être déchiffré !

$pub(E)$: clef publique, la clef privée est partagée entre les autorités.

Trop simplifié !

$\{v_D\}_{\text{pub}(E)}$

Urne

Alice	$\{v_A\}_{\text{pub}(E)}$
Bob	$\{v_B\}_{\text{pub}(E)}$
Chris	$\{v_C\}_{\text{pub}(E)}$
David	$\{v_D\}_{\text{pub}(E)}$
...	...

$v_A = 0$ or 1
$v_B = 0$ or 1
$v_C = 0$ or 1

Résultat : $\{v_A + v_B + v_C + v_D + \cdots\}_{\text{pub}(E)}$

Trop simplifié !

$\{v_D\}_{\text{pub}(E)}$

Urne		
Alice	$\{v_A\}_{\text{pub}(E)}$	$v_A = 0$ or 1
Bob	$\{v_B\}_{\text{pub}(E)}$	$v_B = 0$ or 1
Chris	$\{v_C\}_{\text{pub}(E)}$	$v_C = 0$ or 1
David	$\{v_D\}_{\text{pub}(E)}$	$v_D = 100$
...	...	

Résultat : $\{v_A + v_B + v_C + 100 + \cdots\}_{\text{pub}(E)}$

Un votant pourrait tricher !

Trop simplifié !

$\{v_D\}_{\text{pub}(E)}$

Urne	
Alice	$\{v_A\}_{\text{pub}(E)}$
Bob	$\{v_B\}_{\text{pub}(E)}$
Chris	$\{v_C\}_{\text{pub}(E)}$
David	$\{v_D\}_{\text{pub}(E)}$
...	...

$v_A = 0$ or 1

$v_B = 0$ or 1

$v_C = 0$ or 1

~~$v_D = 100$~~

Résultat : $\{v_A + v_B + v_C + v_D + \cdots\}_{\text{pub}(E)}$

~~Un votant pourrait tricher !~~

Utilisation d'une preuve à connaissance nulle

$$\{v_D\}_{\text{pub}(E)}, \text{SPK}\{v_D = 0 \text{ ou } 1\}$$

Encore trop simplifié

$\{v_D\}_{\mathbf{pub}(E)}$

Urne

Alice	$\{v_A\}_{\mathrm{pub}(E)}$
Bob	$\{v_B\}_{\mathrm{pub}(E)}$
Chris	$\{v_C\}_{\mathrm{pub}(E)}$
...	...
...	

Encore trop simplifié

$\{v_D\}_{\mathbf{pub}(E)}$

Urne	
Alice	$\{v_A\}_{\mathrm{pub}(E)}$
Bob	$\{v_B\}_{\mathrm{pub}(E)}$
Chris	$\{v_C\}_{\mathrm{pub}(E)}$
...	$\{1\}_{\mathrm{pub}(E)}$
...	$\{1\}_{\mathrm{pub}(E)}$

L'urne pourrait ajouter des bulletins !

Encore trop simplifié

$\{v_D\}_{\mathsf{pub}(E)}$

$\mathsf{vk}(cred_3), \mathsf{vk}(cred_1), \mathsf{vk}(cred_2), \ldots$

Urne

Alice	$\{v_A\}_{\mathsf{pub}(E)}$
Bob	$\{v_B\}_{\mathsf{pub}(E)}$
Chris	$\{v_C\}_{\mathsf{pub}(E)}$
...	
...	

~~L'urne pourrait ajouter des bulletins~~ !

Les votants signent avec un "crédit" qu'ils ont reçu (un droit de vote = un crédit).

Encore trop simplifié

$\{v_D\}_{\mathsf{pub}(E)}$

$\mathsf{vk}(cred_3), \mathsf{vk}(cred_1), \mathsf{vk}(cred_2), \ldots$

Urne

Alice	$[\{v_A\}_{\mathsf{pub}(E)}]_{sk(()cred_1)}$
Bob	$[\{v_B\}_{\mathsf{pub}(E)}]_{sk(()cred_2)}$
Chris	$[\{v_C\}_{\mathsf{pub}(E)}]_{sk(()cred_3)}$
...	
...	

~~L'urne pourrait ajouter des bulletins~~ !

Les votants signent avec un "crédit" qu'ils ont reçu (un droit de vote = un crédit).

Sécurité et vérifiabilié de Bélénios

1. ✓ L'urne doit être publique, et le vote vérifiable
2. ✗ Seuls les inscrits votent, et ils sont identifiés à ce moment
3. ✓ Chacun peut vérifier que les votes viennent d'inscrits, et que chacun n'a voté qu'une fois (pas de bourrage)
4. ✓ Tous doivent pouvoir vérifier le comptage final
5. ✓ Secret : l'urne ne doit pas permettre de savoir qui a voté quoi
6. ✓ Chaque votant doit pouvoir vérifier que son vote est dans l'urne et pas modifié
7. ✗ Mais il ne doit pas pouvoir prouver à son vote à un tiers (le vendre)
8. ? Et tout cela même si les autorités sont malhonnêtes

Problème : on ne sait pas faire tout ça en même temps…
(la CNIL parle de secret, mais pas de vérifiabilité !)

Limitations de Belenios

- Pas de réel isoloir
  → Le vote par Internet EST du vote par correspondance
- Il faut faire confiance à l'ordinateur
  Un ordinateur infecté pourrait :
  - faire fuiter le choix du votant
  - voter pour un candidat différent
    → exemple de l'attaque de Laurent Grégoire lors des "législatives 2012, pour les français de l'étranger" (injection de code).
- Belenios n'est pas "sans reçu"
  → On peut prouver pour qui on a voté.

Agenda

1. Les acteurs et leurs points de vue
2. Les questions de sécurité – exemples d'attaques
3. Quelles actions pour les chercheurs ?
4. Chiffrement et signature
5. Les protocoles de sécurité
6. Un bon exemple : le vote « électronique »
7. Les microprocesseurs aussi sont atteints !

Meltdown et *Spectre*
Attaquer la microarchitecure des processeurs

- Attaques redoutables découvertes en 2018 et toujours actives sur la plupart des processeurs et OS modernes

Meltdown: Reading Kernel Memory from User Space

Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas,,ers Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, Mike Hamburg

USENIX Security Symposium 2018

https://meltdownattack.com/meltdown.pdf

Spectre Attacks: Exploiting Speculative Execution

Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, Yuval Yarom

40th IEEE Symposium on Security, Privacy, 2019

https://spectreattack.com/spectre.pdf

De l'architecture à la microarchitecture

Architecture : définie par l'ISA, *Instruction Set Architecture*

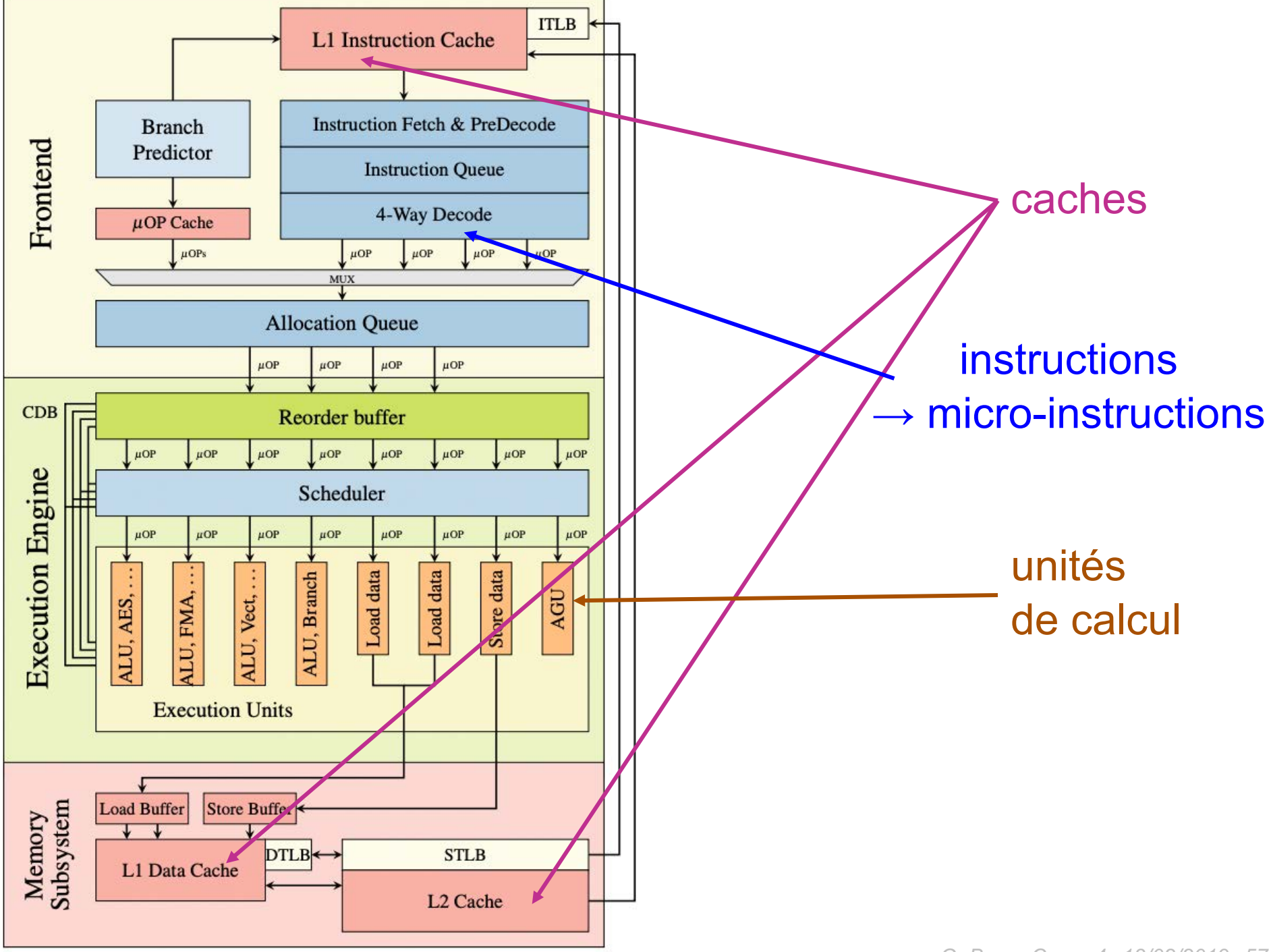
définit les registres et le résultat de l'exécution des instructions
beaucoup d'instructions (+1000 pour Pentium), gros livres !

Microarchitecture : la mécanique d'exécution

autrefois proche de l'architecture (Motorola 68000 p. ex.)
maintenant très éloignée : nombreux mécanismes d'optimisation

raisons : des opérations sont plus lentes que le cycle d'horloge
la mémoire est beaucoup plus lente que le processeur
si on ne fait rien le processeur passe sa vie à attendre !

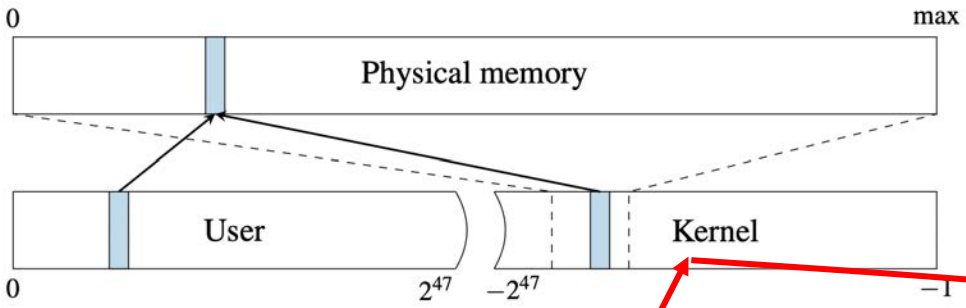
microinstructions, pipelines
caches (tampons rapides entre RAM et processeurs)
exécution hors d'ordre (annulables)
prédiction de tests, exécution spéculatives de branches (idem)
⇒ jeux de registres multiples (*shadowing*)


Frontend
L1 Instruction Cache
ITLB
Branch Predictor
Instruction Fetch & PreDecode
Instruction Queue
4-Way Decode
µOP Cache
µOPs
µOP
MUX
Allocation Queue
Execution Engine
CDB
Reorder buffer
Scheduler
ALU, AES, ...
ALU, FMA, ...
ALU, Vect, ...
ALU, Branch
Load data
Load data
Store data
AGU
Execution Units
Memory Subsystem
Load Buffer
Store Buffer
L1 Data Cache
DTLB
STLB
L2 Cache
caches
instructions
→ micro-instructions
unités
de calcul

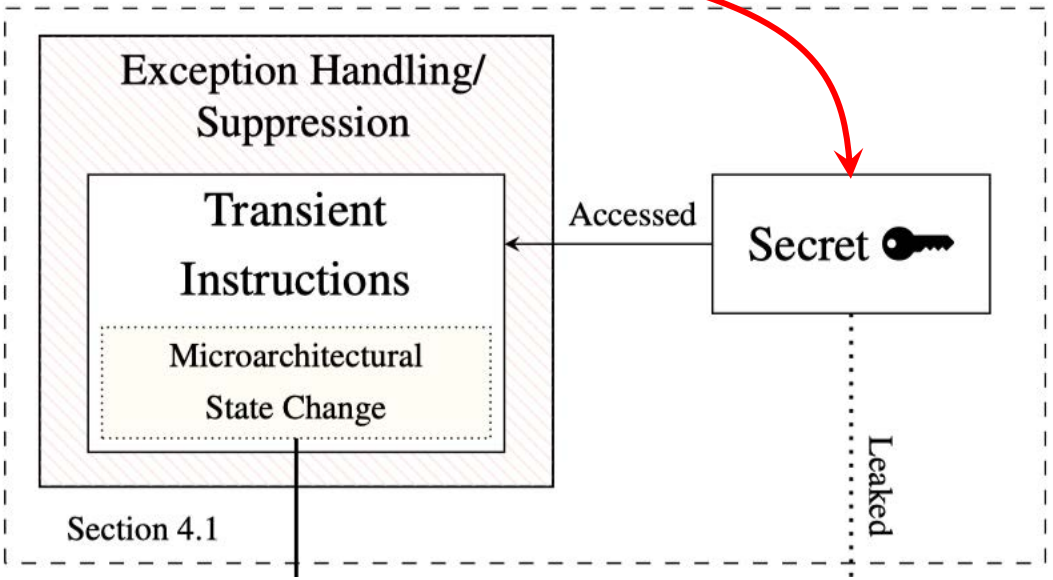
Meltdown, phase 1 : exposer la mémoire en cache

- Le processeur permet la mémoire virtuelle. En particulier, l'OS peut mapper toute de la mémoire physique dans sa mémoire noyau. Il est architecturalement interdit d'y accéder directement (*illegal memory read*), mais c'est microarchitecturalement possible.
- Une instruction calculée spéculativement peut lire la mémoire virtuelle n'importe où, ce qui lui fait mettre les valeurs lues dans le cache, et calculer avec des registres microarchitecturaux (*shadows*).
- Si l'instruction est finalement exécutée, les *shadows* sont transférés dans les vrais registres et l'instruction terminée (*retired*).
- Sinon, les *shadows* sont simplement ignorés, mais le cache contient toujours les valeurs lues !

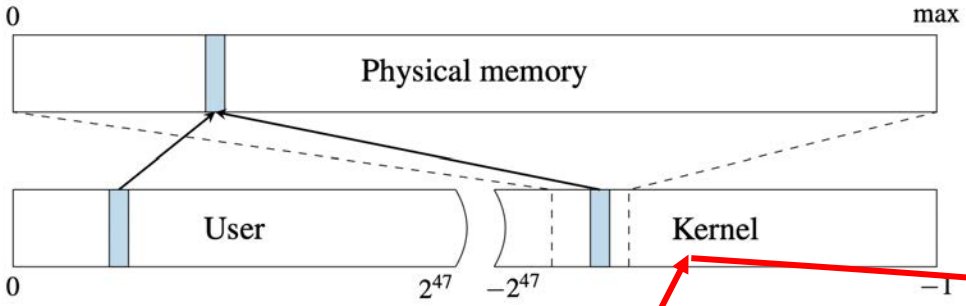
Donc, en balayant les adresses physiques, on peut mettre une par une toutes les pages de la mémoire en cache



OK inst1 ;
OK inst2 ;
TRAP [exception] ;
Specul. ~~access(A[data * 4096])~~ ;



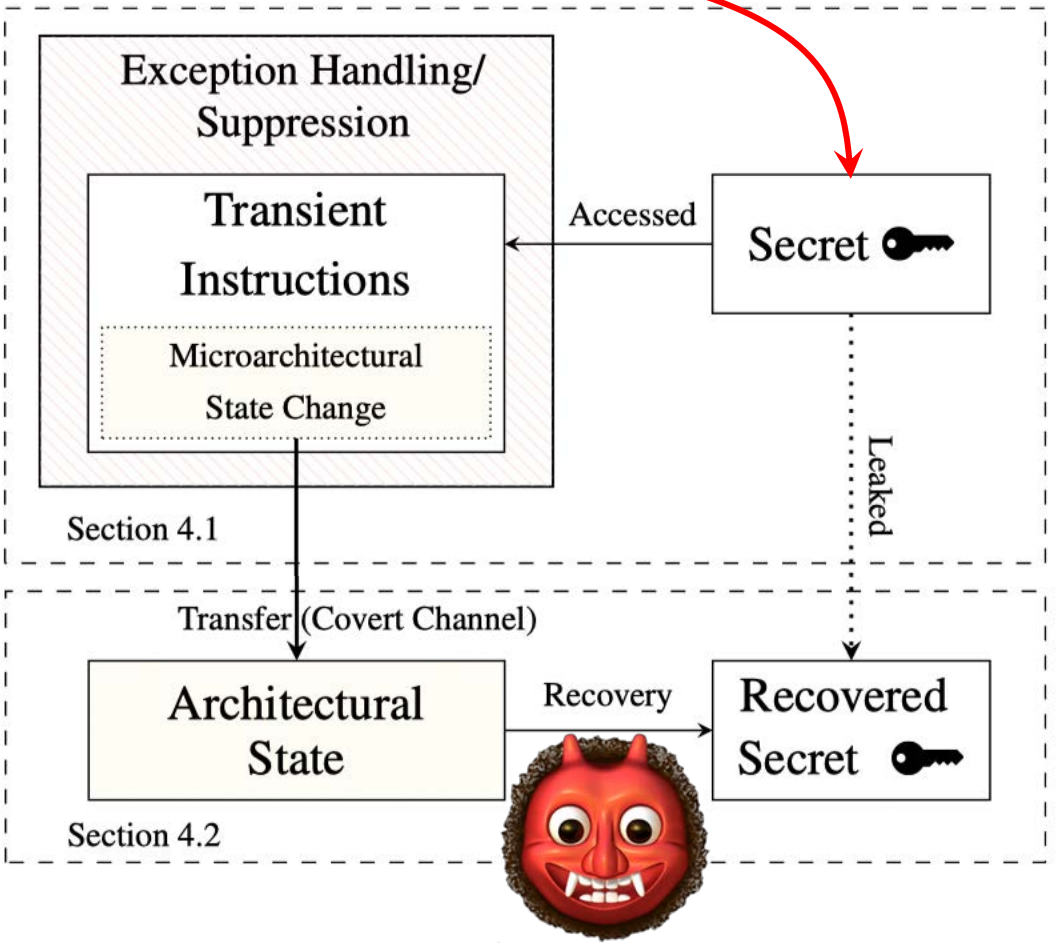
Le secret est arrivé dans la microarchitecture



Phase 2 :

μ-arch → arch

OK inst1 ;
OK inst2 ;
TRAP [exception] ;
Specul. ~~access(A[data * 4096])~~ ;



flush & reload

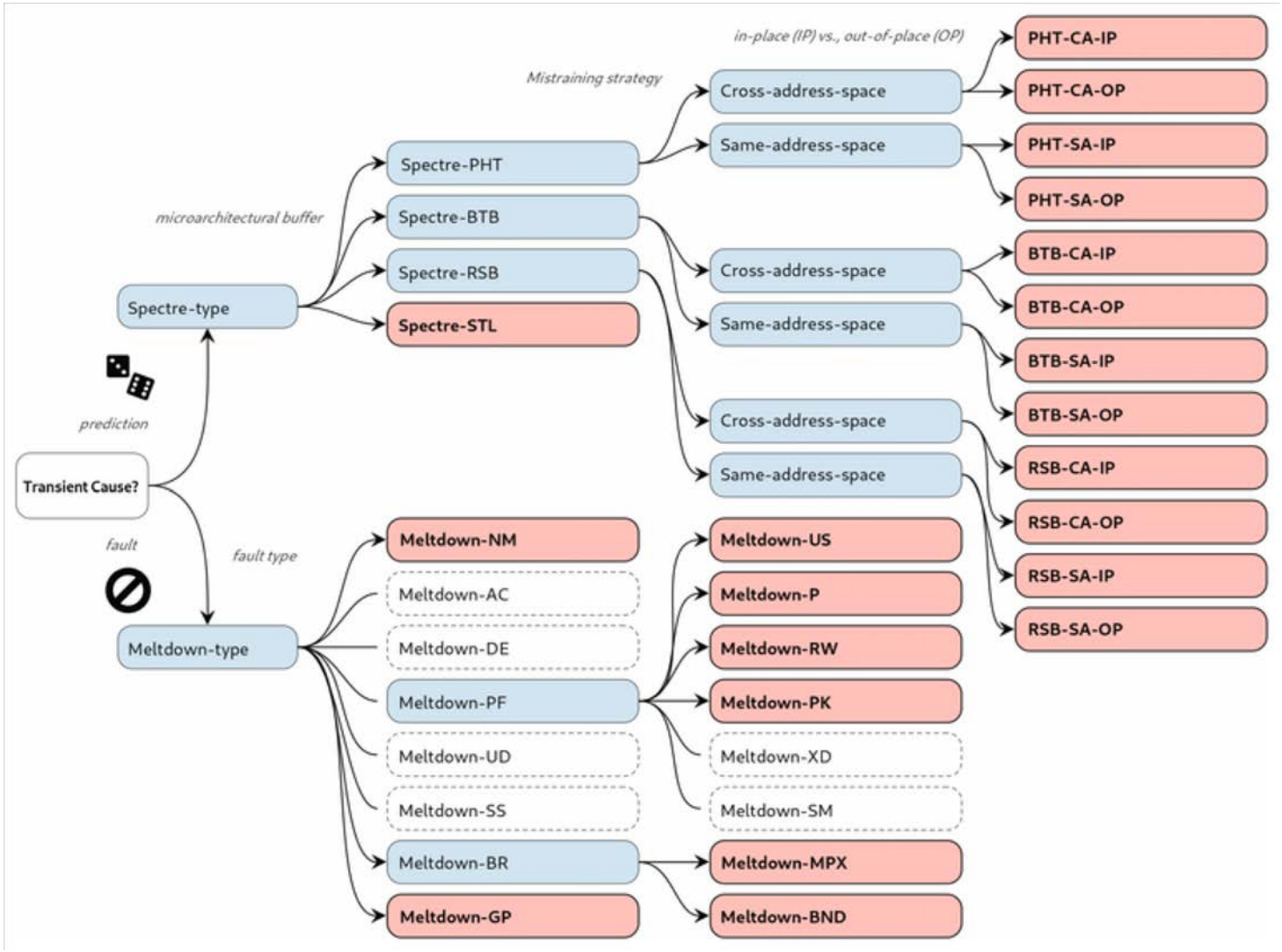
si accès rapide, alors
la valeur est en cache

timing covert channel

Spectre : attaque de la spéculation

- if Cond then P else Q : si Cond est long à calculer, on perd beaucoup de temps. Solution : essayer de prédire la branche qui va être exécutée, et l'exécuter spéculativement. Pour aller vite, la prédiction se fonde sur des heuristiques variées.
- Si un utilisateur fait tourner un programme R dont on connaît le code, par exemple sur un serveur, on peut mettre sur ce serveur un autre programme X qui va entraîner certaines prédictions à prendre la mauvaise branche.
- Ces mauvaises branches vont laisser des traces dans le cache, récupérables architecturalement comme pour Meltwown

Beaucoup de variantes !



Certaines corrigées par des patchs soft ou μ-soft, d'autres non…

Meltdown pas mal corrigé, mais nouvelles variantes

Spectre plus dur !

Table 7: Spectre-type attacks and whether a defense mitigates it.

	Attack \ Defense	InvisiSpec [88]	SafeSpec [45]	DAWG [47]	RSB Stuffing [39]	Retpoline [82]	Poison Value [82]	Index Masking [69]	Site Isolation [69]	SLH [12, 17]	YSNB [65]	IBRS [4, 40]	STIPB [4, 40]	IBPB [4, 40]	Serialization [1, 37]	Taint Tracking [50]	Timer Reduction [50]	Sloth [48]	SSBD/SSBB [3, 40, 6]
Intel	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	●	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	●	◐	◐	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●
ARM	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	◐	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●
AMD	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	◐	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	■	◧	◧	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇). Empty fields still require testing.

A Systematic Evaluation of Transient Execution Attacks and Defenses
Claudio Canella, Jo Van Bulck, Michael Schwarz, Moritz Lipp, Benjamin von Berg, Philipp Ortner, Frank Piessens, Dmitry Evtyushkin, Daniel Gruss
arXiv 2018

Conclusion

- Les attaques contre les systèmes et données deviennent majeures, avec des impacts déjà ravageurs et probablement destinés à augmenter
- Elles mettent en question la viabilité de l'Internet des objets
- Mais le problème n'est pas encore considéré suffisamment sérieusement dans beaucoup d'organisations, ni chez le citoyen moyen (qui ne sait pas où se renseigner)
- Beaucoup de vulnérabilités sont dues à des bugs classiques, p.ex. mauvais tests d'accès mémoire – efficacité oblige !
- Les chercheurs progressent, mais les pirates aussi….

Pardonnez mes erreurs probables, je ne suis pas spécialiste !
(mais je tenais à en parler dans mon dernier cours technique…