

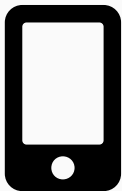
Parcours :

- 2013 - 2017 ; Licence et Master à l'ENS Cachan
- Sep. 2017 - Sep. 2020 ; Thèse au LSV et LORIA avec Hubert Comon et Steve Kremer
- Nov. 2020 - 2022 ; Postdoc au CISPA (Allemagne) avec Cas Cremers

Méthodes formelles en sécurité

Charlie Jacomme

16 Mars 2022



La révolution technologique



Le droit à la vie privée est **vital** pour certaines personnes :

- Les journalistes dans des endroits dangereux.
- Les homosexuels dans des pays où c'est un crime. (encore 69 dans le monde...).
- Les Ouïghours traqués via leurs téléphones en Chine.

Même si on n'a rien à cacher, c'est important de le cacher.

La sécurité de nos données et le droit à la vie privée sont essentiels !

L'un des problèmes scientifiques

Comment fournir des garanties **formelles**, i.e., des preuves de sécurité ?

Protocoles



SSH

TLS

GPG

...

La première difficulté

Matériel

OS

Implémentations

Primitives

Protocoles

Utilisateurs



C++
Java
Python
...

RSA
AES
...

SSH
TLS
GPG
...

La première difficulté

Matériel

OS

Implémentations

Primitives

Protocoles

Utilisateurs



C++
Java
Python
...

RSA
AES
...

SSH
TLS
GPG
...

La première difficulté

Matériel

OS

Implémentations

Primitives

Protocoles

Utilisateurs



C++
Java
Python
...



RSA
AES
...



SSH
TLS
GPG
...



L'objectif

Depuis les années 80

Obtenir des garanties **formelles** sur la sécurité du protocole en supposant les autres niveaux sécurisés.

↪ une preuve mathématique dans un modèle [Goldwasser,Micali,Dolev,Yao]

Preuve de protocole

En supposant que RSA est sécurisé, prouver dans un modèle qu'**aucun attaquant** ne peut casser la sécurité du protocole SSH.

Modèles d'attaquant

Modèles d'attaquant

Modèles de calcul

- Machine de Turing vs règles d'inférence
- Hypothèses sur les primitives (RSA)

Modèles d'attaquant

Modèles de calcul

- Machine de Turing vs règles d'inférence
- Hypothèses sur les primitives (RSA)

Modèles de compromissions

- Virus, Keylogger
- Phishing
- Clé long-terme ou éphémère

Modèles d'attaquant

Modèles de calcul

- Machine de Turing vs règles d'inférence
- Hypothèses sur les primitives (RSA)

Modèles de compromissions

- Virus, Keylogger
- Phishing
- Clé long-terme ou éphémère

Modèles du protocole

- Comportements optionnels
- Granularité des appels de fonctions

Deuxième difficulté - La modélisation

Modèles d'attaquant

Modèles de calcul

- Machine de Turing vs règles d'inférence
- Hypothèses sur les primitives (RSA)

Modèles de compromissions

- Virus, Keylogger
- Phishing
- Clé long-terme ou éphémère

Propriétés de sécurité

- Secret (robuste)
- Authentification
- Vie privée

Modèles du protocole

- Comportements optionnels
- Granularité des appels de fonctions

Des garanties fortes

Obtenir des preuves de sécurité dans des modèles aussi **réalistes** que possible.

↪ La complexité des preuves est vite trop élevée.

L'objectif

Des garanties fortes

Obtenir des preuves de sécurité dans des modèles aussi **réalistes** que possible.

↪ La complexité des preuves est vite trop élevée.

Cryptographie assistée par ordinateur (depuis les années 2000)

Des programmes nous aident à **faire, vérifier ou automatiser** les preuves.

(Proverif, Tamarin, Deepsec, EasyCrypt, CryptoVerif, Squirrel...)

Autour des protocoles

- Une analyse **détaillée** via Proverif des protocoles d'authentification multi-facteurs.
[Kremer, **Jacomme** - **CSF'18** (A), **ACM TOPS** (A)]
- Un résultat de composition pour couper les preuves en des morceaux **modulaires**.
[Comon, **Jacomme**, Scerri - **CCS'20** (A*)]
- **Squirrel**, un assistant de preuve pour la sécurité de protocoles.
[Baelde, Delaune, **Jacomme**, Koutsos, Moreau - **S&P'21** (A*)]
- Extension de Squirrel à des attaquants **quantiques**.
[Cremers, Fontaine, **Jacomme** - **S&P'22** (A*)] **Nouvelle publication du postdoc**

Autour de la preuve de programme et des primitives

- Décision de la connaissance de l'attaquant sur les **groupes finis**.
[Barthe, Fan, Ganher, Grégoire, **Jacomme**, Shi - **CCS'18** (A*)]
- Algorithmes de décision pour l'**égalité de distributions** et de propriétés de non-interférence pour les primitives.
[Barthe, Grégoire, **Jacomme**, Kremer, Strub - **CSF'19** (A)]
- Complexité et décidabilité de l'**équivalence de programmes** probabilistes sur des corps finis.
[Barthe, **Jacomme**, Kremer - **LICS'20** (A*), **ACM TOCL** (A)]

Développements théoriques

- Travaux théoriques à l'intersection de **plusieurs domaines**.
Sécurité Symbolique, Sécurité Calculatoire, Complexité, Décidabilité, Probabilités, Géométrie algébrique, Logique, Preuve de programme, Calcul Quantique

Mise en pratique

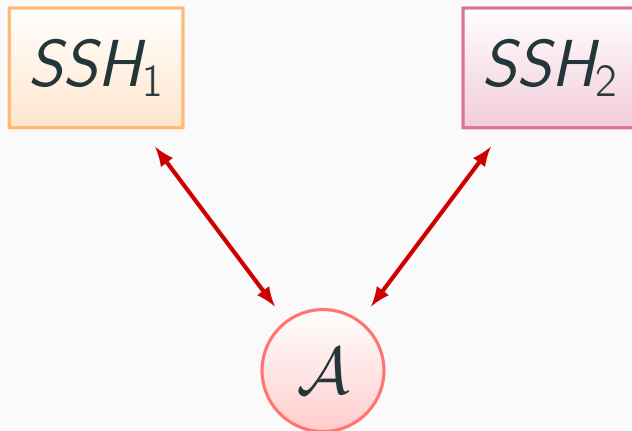
- Utilisation et/ou amélioration de **plusieurs outils** du domaine.
Tamarin, Proverif, DeepSec, Sapic, EasyCrypt, MaskVerif, AutoGnP
- Le développement **logiciel** de Squirrel¹ (~ 30 000 lignes de code)
- Plusieurs **études de cas** (preuves de sécurité et détection de failles) :
 - Analyse de 6 000 scénarios de protocoles d'authentification multi-facteurs ;
 - preuve en Squirrel de SSH et de protocoles d'échange de clés post-quantiques.

¹squirrel-prover.github.io

Composition

Composition de deux protocoles

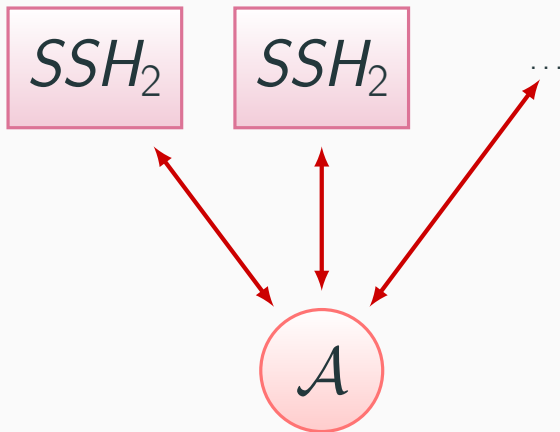
Si on a une preuve de sécurité pour SSH_1 contre un attaquant \mathcal{A} et une preuve de sécurité pour SSH_2 contre \mathcal{A} , a-t-on une preuve des deux en même temps ?



La composition

D'une seule session à un nombre non borné

Si on a une preuve de sécurité pour une session de SSH_2 , a-t-on une preuve pour plusieurs en même temps ?



Limitations de l'état de l'art

Un sujet très étudié depuis plus de 20 ans, mais même les papiers récents ont toujours des limitations :

- Ne s'applique qu'à des cas de compositions restreints et précis ;

Blanchet, CSF'18

- Ne s'utilise que très difficilement avec les outils ;

Camenisch et al., ASIACRYPT'19

- Ne gère pas les protocoles avec passage d'état ou partage de secrets long terme, et donc pas la composition d'une session avec un nombre non borné.

Brzuska et al., ASIACRYPT'18

Une technique générale de composition

- Un résultat générique de composition supportant le passage d'état, le partage de secrets long terme, et la réplication non bornée.
- Technique utilisable dans de nombreux outils existants.
(e.g., Tamarin, ProVerif, DeepSec, Squirrel, CryptVerif, EasyCrypt)
- Preuve de SSH en Squirrel.

Oracle simulation: a technique for protocol composition with long term shared secrets.
*H. Comon, C. **Jacomme** and G. Scerri - CCS'20*

Simulation par oracle

On donne à l'attaquant un accès partiel à la clé secrète via un oracle qui:

- est assez **fort** pour permettre de simuler SSH_1 ;
- mais est assez **faible** pour que SSH_2 soit toujours sécurisé.

Dans ce cas précis

Tous les messages de SSH_x contiennent un tag "*version x*".

↪ On peut prouver la sécurité de SSH_2 pour un attaquant $\mathcal{A}^{\mathcal{O}}$, où \mathcal{O} permet d'utiliser la clé secrète pour tout message ne contenant pas "*version 2*".

Théorème (informel)

Si \mathcal{O} permet de simuler \mathcal{Q} , alors :

$$\begin{aligned} \forall \mathcal{A}. \mathcal{P} \parallel \mathcal{A}^{\mathcal{O}} \models \phi \\ \Rightarrow \\ \forall \mathcal{A}. \mathcal{P} \parallel \mathcal{Q} \parallel \mathcal{A} \models \phi \end{aligned}$$

De plusieurs sessions à une seule

Un problème plus profond

Deux copies de SSH_2 ont exactement **le même comportement** et utilisent le même code.

Ce qui nous sauve

Chaque copie de SSH_2 doit tirer de l'aléa frais au moment de son exécution.

Renversement conceptuel

- Traiter tous les **aléa frais comme des secrets long terme** tirés magiquement au début de l'univers plutôt qu'au moment de l'exécution du protocole ;
- Donner un oracle à l'attaquant lui permettant d'utiliser la clé secrète pour tout message qui ne dépend pas de l'aléa frais.

Et après ?

Garanties formelles

Fournir des garanties **formelles** aussi **précises** que possible sur tous les protocoles déployés à grande échelle.

↪ demande encore de nombreux développements théoriques et pratiques

Preuves d'échanges de clés

Squirrel et le résultat de composition permettent déjà d'aller plus loin que les autres outils, mais on a uniquement regardé des propriétés simples pour le moment.

Fondations pour les preuves de protocoles d'échange de clés en Squirrel

- Définir en Squirrel des **propriétés complexes** de secret robuste, simplifiées avec la composition.
- Lier **formellement** les preuves en Squirrel avec les techniques de preuve classiques pour ces protocoles (BR, CK, eCK, ...).
- Valider par des études de cas (Signal, KEMTLS, ...).

Faire des analyses multi-niveaux

Pour faire une analyse approfondie d'un protocole, il est nécessaire de combiner les avantages de plusieurs outils, ce qui est peu fait aujourd'hui.

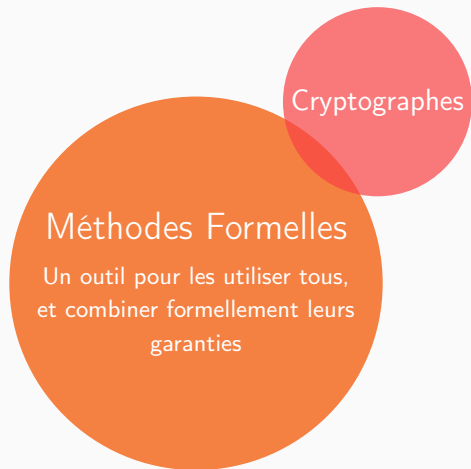
Inter-opérabilité

Une plateforme qui permet de combiner **automatiquement** et **formellement** les garanties de plusieurs outils :

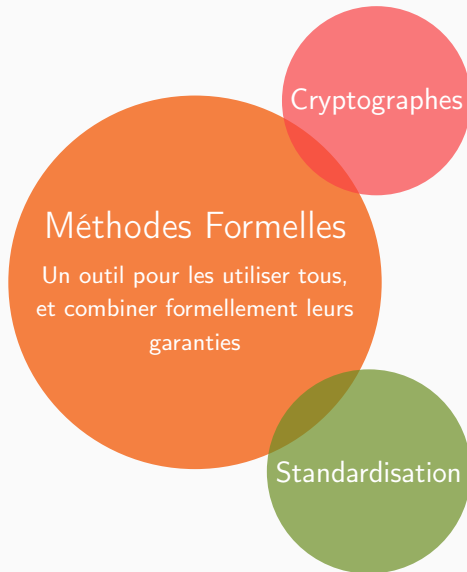
- D'abord pour Proverif et Tamarin, pour combiner leurs forces respectives ;
- puis intégrer Squirrel, CryptoVerif, ... ;
- Et faire des études de cas de manière exhaustive, à plusieurs niveaux de détail et avec plusieurs outils.

Méthodes Formelles

Un outil pour les utiliser tous,
et combiner formellement leurs
garanties



- Utiliser les outils dès le design des nouveaux protocoles.



- Utiliser les outils dès le design des nouveaux protocoles.

- Vérifier les standards existants ;
- Participer au développement des nouveaux standards.

- Rendre accessibles à des ingénieurs certaines de nos techniques.



- Utiliser les outils dès le design des nouveaux protocoles.

- Vérifier les standards existants ;
- Participer au développement des nouveaux standards.

Sur le long terme

- Diverses opportunités via l'intégration dans un laboratoire

Par delà les protocoles ?

Cryptographes

- Utiliser les outils dès le design des nouveaux protocoles.

Méthodes Formelles

Un outil pour les utiliser tous,
et combiner formellement leurs
garanties

- Rendre accessibles à des ingénieurs certaines de nos techniques.

Entreprises
Gouvernements

Standardisation

- Vérifier les standards existants ;
- Participer au développement des nouveaux standards.

IRISA - Rennes

- Insertion naturelle avec collaborations existantes (D. Baelde et S. Delaune)
- + Implication dans les standards (Mohamed Sabt).
- + Preuves de sécurité de systèmes de base de données (Tristan Allard).

LIMOS - Clermont

- Insertion naturelle avec Pascal Lafourcade sur les protocoles.
- + Preuves de sécurité de systèmes industriels (également Pascal Lafourcade).

VERIMAG - Grenoble

Ouverture thématique vers la preuve de programme :

- modèles d'attaquant au niveau du code pour l'injection de fautes ou les processeurs sécurisés (Cristian Ene, David Moniaux, Marie-Laure Potet).

Parcours

- 2013 - 2017 ; Licence et Master à l'ENS Cachan.
- Sep. 2017 - Sep. 2020 ; Thèse au LSV et LORIA avec Hubert Comon et Steve Kremer.
- Nov. 2020 - 2022 ; Postdoc au CISPA (Allemagne) avec Cas Cremers.

Résumé du CV académique et nouveautés

- Prix de thèse du **GdR sécurité** et **accessit Gilles Kahn**
- Conférences = 4 A*, 2 A, 1 New ; Journaux = 2 A
+ 1 nouvelle publication à **S&P'22** (A*) + 1 sheperding à **USENIX'22** (A*).
- Collaborations à Rennes, Paris, Nancy, Nice, Bochum, New-York et Saarbrucken.
- **Comité de Programme** de INDOCRYPT'21 (B) et de CSF'22 (A, *principale de la communauté méthodes formelles*)
- Implication dans la standardisation à l'**IETF** d'un protocole.
(<https://datatracker.ietf.org/doc/agenda-113-lake/>)

Nouvelles publications depuis la soumission du dossier

S&P'22

Extension de Squirrel à des attaquants **quantiques**.

A Logic and an Interactive Prover for the Computational Post-Quantum Security of Protocols. Cremers, Fontaine, **Jacomme** - **S&P'22** (A*)

USENIX'22 - sheperding

Une plateforme pour utiliser Proverif et Tamarin en combinaison.

SAPIC+: protocol verifiers of the world, unite! Cheval, **Jacomme**, Kremer, Künnemann - **USENIX'22** (A*)