A first glance at the Universal Second Factor (U2F) protocol

Florian Maury
ANSSI

Mickaël Bergem
ParisTech

2 juin 2016
SSTIC

Pourquoi faire un nouveau protocole d'authentification à deux facteurs ?

Des problèmes avec les codes SMS et TOTP

Retransmission des codes :

- SMS et TOTP vulnérables au hameçonnage

Secrets symétriques :

- stockage en clair d'information par le serveur

Les problèmes des codes envoyés par SMS

SS7

FIGURE : https://xkcd.com/525/

SS7 est peu sûr (SS7 Security Report – Positive Technologies)

Universal Second Factor (U2F)

U2F : qui ? quand ? quoi ? comment ?

Qui ?

- normalisation : consortium FIDO Alliance
- services actuellement protégeables par U2F : Google, Github, Dropbox

Quand ?

- spécifications 1.0 : mai 2015

Quoi ?

- tokens cryptographiques logiciels, USB, BT ou NFC

Comment ?

- tokens disponibles en ligne

U2F : qui ? quand ? quoi ? comment ?

Qui ?

- normalisa ce
- services a U2F : Google, Github, D

Quand ?

- spécificat

Quoi ?

- tokens cry es lo u NFC

Comment ?

- tokens disponibles en ligne

U2F : qui ? quand ? quoi ? comment ?

Qui ?

- normalisation : consortium FIDO Alliance
- services actuellement protégeables par U2F : Google, Github, Dropbox

Quand ?

- spécifications 1.0 : mai 2015

Quoi ?

- tokens cryptographiques logiciels, USB, BT ou NFC

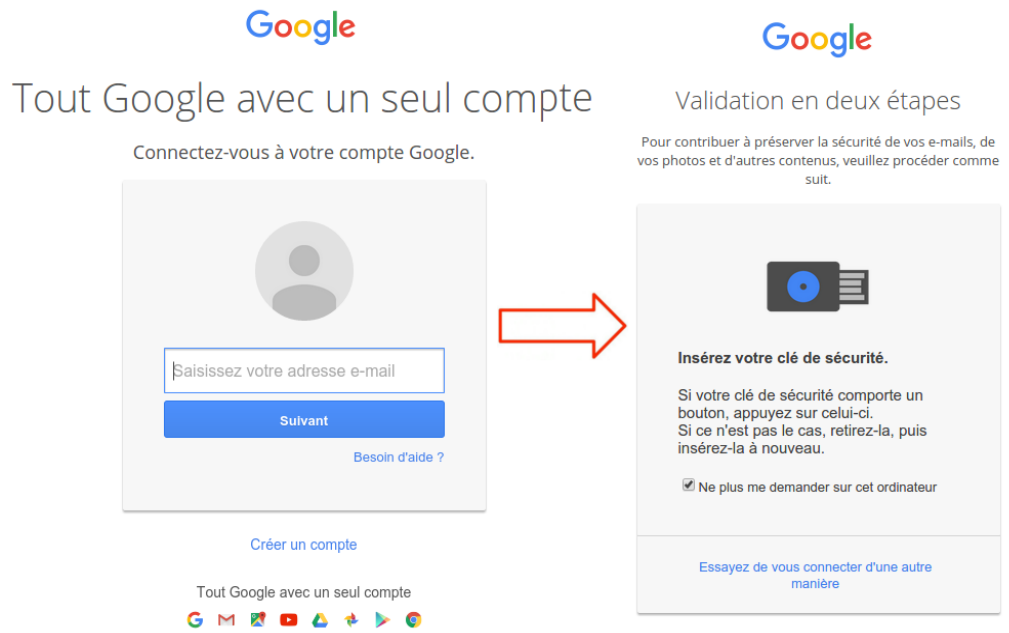
Comment ?

- tokens disponibles en ligne

Procédure d'authentification U2F typique

Authentification en deux étapes :

- saisie d'un mot de passe
- puis, activation du token U2F



Aperçu de la cryptographie d'U2F

Cryptographie asymétrique :

- une bi-clé par *origin*
- ECDSA sur P-256
- validation manuelle unitaire

Aperçu de la cryptographie d'U2F

Cryptographie asymétrique :

- Une *origin* = (scheme, host, port)
  - exemple : (http, www.ssi.gouv.fr, 80)
- validation manuelle unitaire

Aperçu de la cryptographie d'U2F

Cryptographie asymétrique :

- une bi-clé par *origin* $\Leftarrow$ **prévient le hameçonnage**
- ECDSA sur P-256
- validation manuelle unitaire

Aperçu de la cryptographie d'U2F

Cryptographie asymétrique :

- une bi-clé par *origin* $\Leftarrow$ **prévient le hameçonnage**
- ECDSA sur P-256
- validation manuelle unitaire $\Leftarrow$ **limite les oracles**

Du stockage des clés cryptographiques

Utilisation d'un identifiant de clé, associé à une *origin*

- *key handle*

Deux types de stockage de clés :

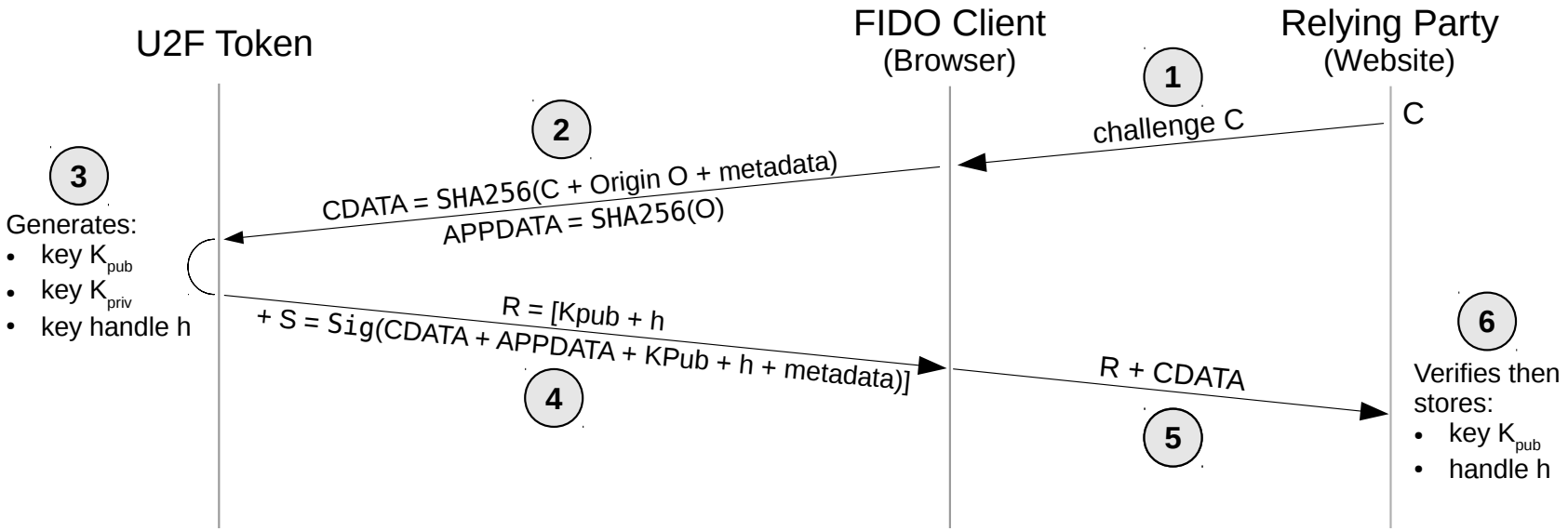
- dans le token ; mémoire interne finie
- hors du token : *key handle* = clé privée *wrappée*

Objectif de la première phase d'authentification :

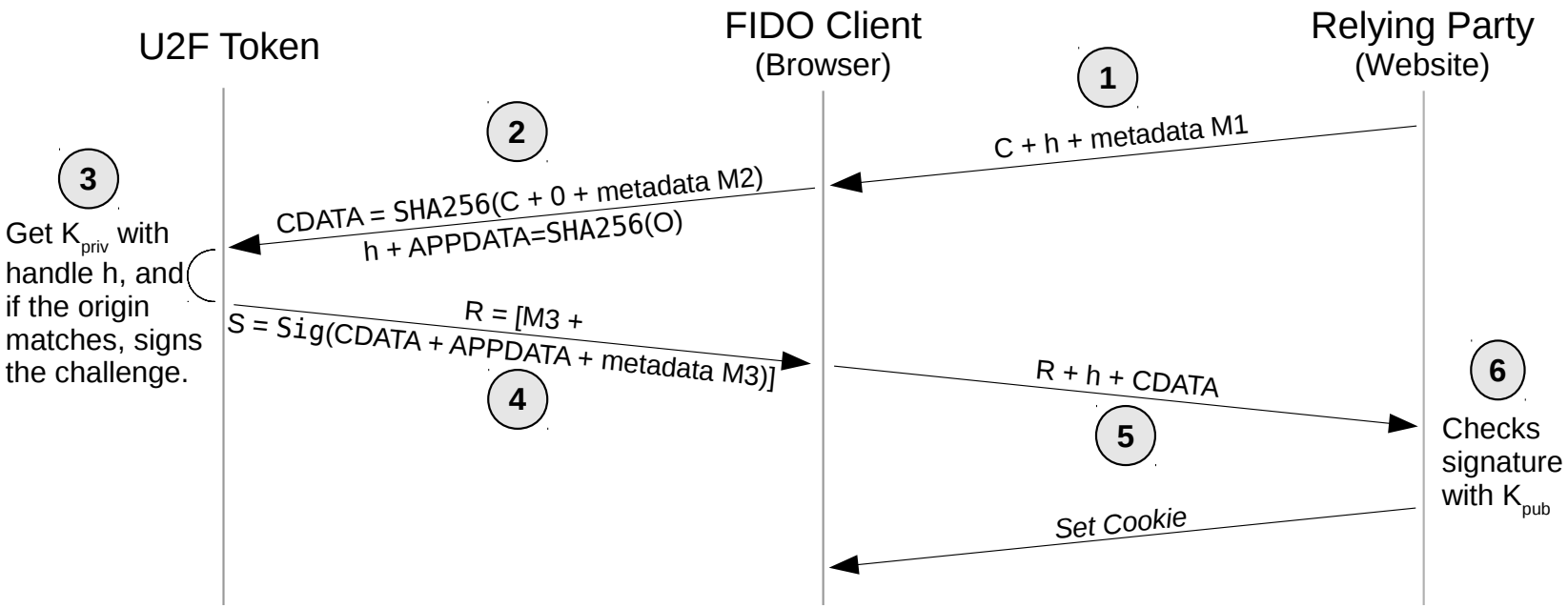
- identifier le *key handle* à utiliser par U2F

Détails des échanges protocolaires

Détails des échanges protocolaires : Enregistrement



Détails des échanges protocolaires : Authentification



Analyse de sécurité

Propriétés de sécurité « étudiées »

Propriétés étudiées :

- anti-rejeu
- résistance au hameçonnage

Prérequis :

- TLS pour assurer la confidentialité du matériel authentifiant
- Chrom{e,ium} 41+

Propriétés de sécurité « étudiées »

Propriétés étudiées :

- anti-rejeu $\Leftarrow$ **nonces, compteur**
- résistance au hameçonnage

Prérequis :

- TLS pour assurer la confidentialité du matériel authentifiant
- Chrom{e,ium} 41+

Propriétés de sécurité « étudiées »

Propriétés étudiées :

- anti-rejeu $\Leftarrow$ **nonces, compteur**
- résistance au hameçonnage $\Leftarrow$ **clé unique par origin**

Prérequis :

- TLS pour assurer la confidentialité du matériel authentifiant
- Chrom{e,ium} 41+

Propriétés de sécurité « étudiées »

Propriétés étudiées :

- anti-rejeu ⇐ **nonces, compteur**
- **n**

Prére

- TLS pour assurer la confidentialité du matériel authentifiant
- Chrom{e,ium} 41+

Avertissement :

- pas de vérification formelle

Contre-mesure aux MITM TLS

Définition du MITM TLS

Dans le cadre d'U2F, MITM TLS signifie :

- attaquant sur le chemin réseau
- même *origin* que le site légitime
- certificat « validé » (par le navigateur ou l'utilisateur)

Présentation de TLS Channel ID

TLS Channel ID :

- extension TLS
- défini dans un Internet-Draft expiré
- redéfinition en cours par le GT tokbind
- prise en charge optionnelle dans U2F
- une bi-clé générée une fois par serveur

Objectif de TLS Channel ID

Être un identifiant liant les deux terminateurs de connexion et le transcript TLS

Objectif de TLS Channel ID

Être un identifiant liant les deux terminateurs de connexion et le transcript TLS

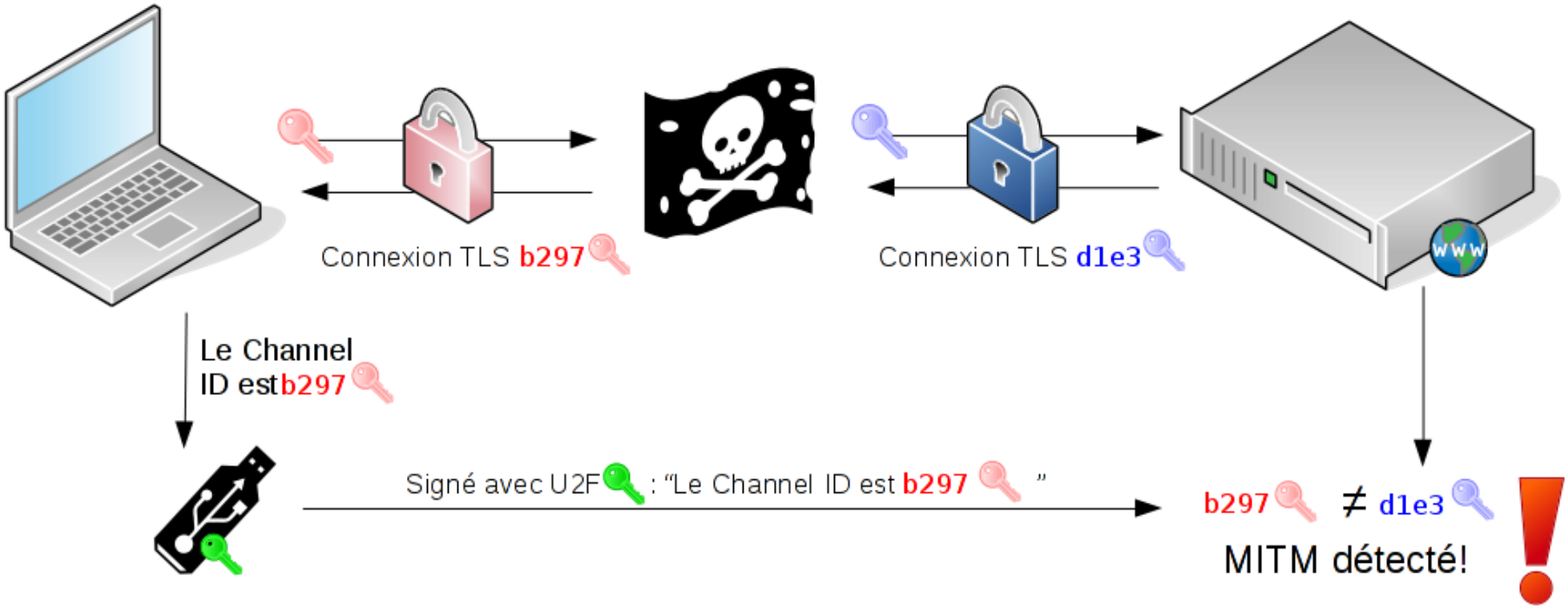


FIGURE : U2F signe le TLS Channel ID pour prévenir le MITM TLS

Objectif de TLS Channel ID

Être un identifiant liant les deux terminateurs de connexion et le transcript TLS

FIGURE : TLS Channel ID est vulnérable aux attaques par dégradation du niveau de sécurité s'il est optionnel et utilisé seul

Objectif de TLS Channel ID

Être un identifiant liant les deux terminateurs de connexion et le transcript TLS

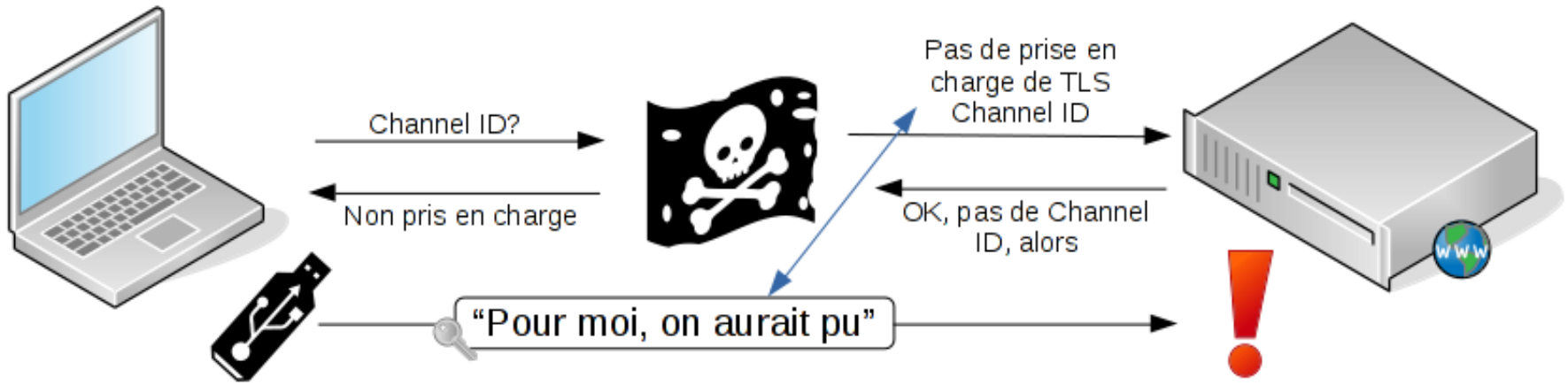


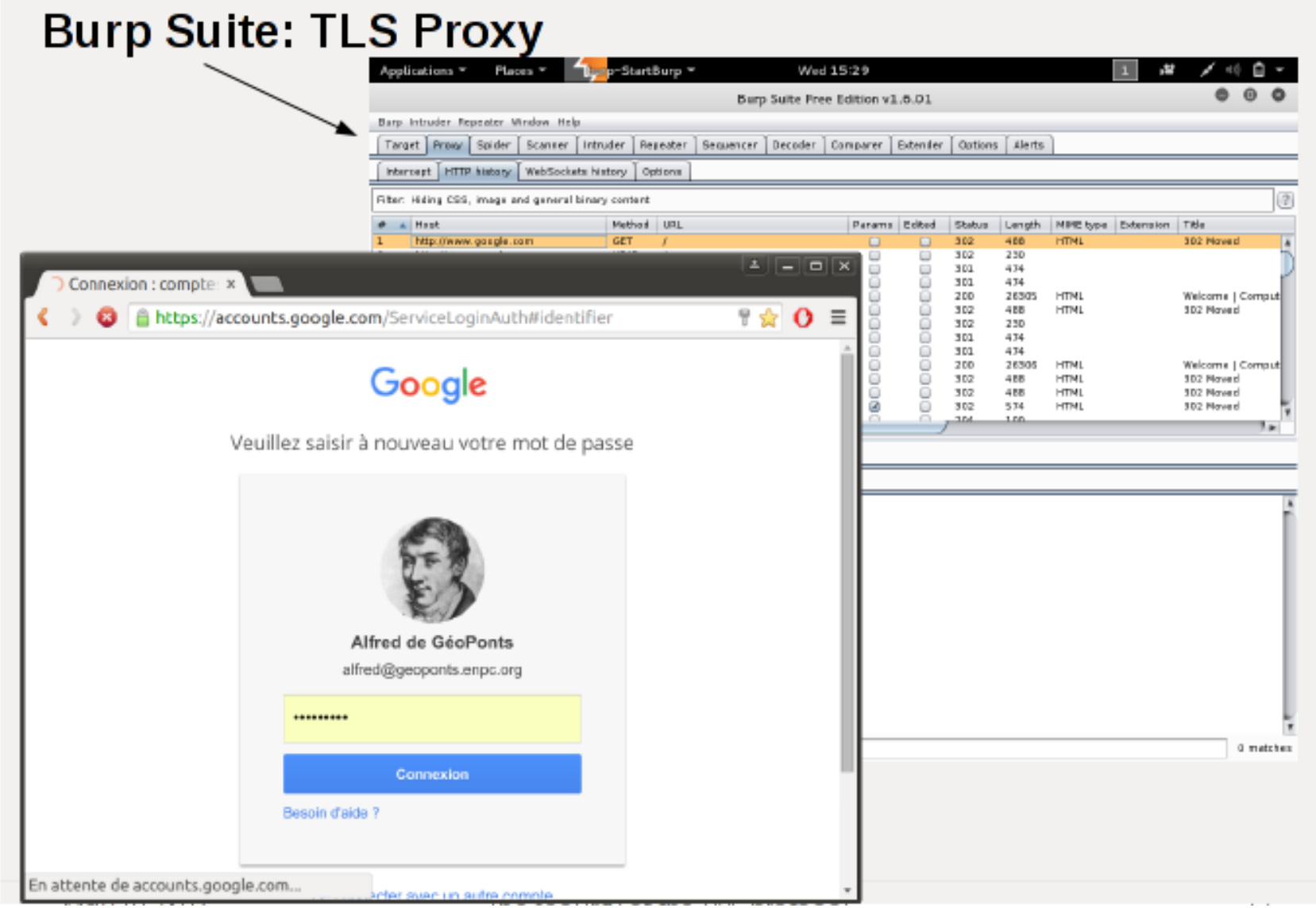
FIGURE : U2F renforce TLS Channel ID en prévenant ce type d'attaque

« Et ça marche, ce truc ? »

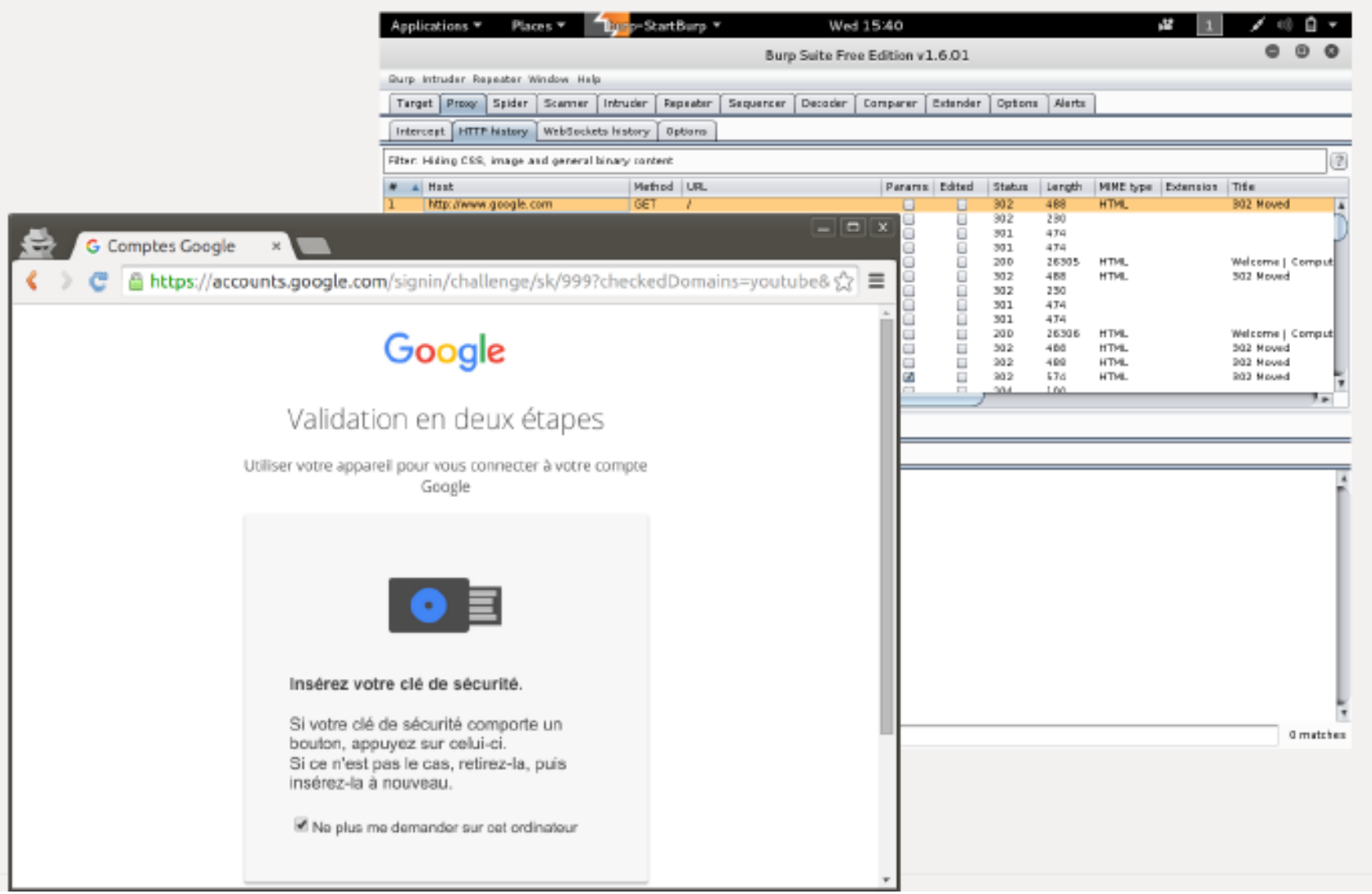
Qui essaie de négocier TLS Channel ID ?

Service Web	Prise en charge de TLS Channel ID
Google	✔
Github	✘
Dropbox	✘

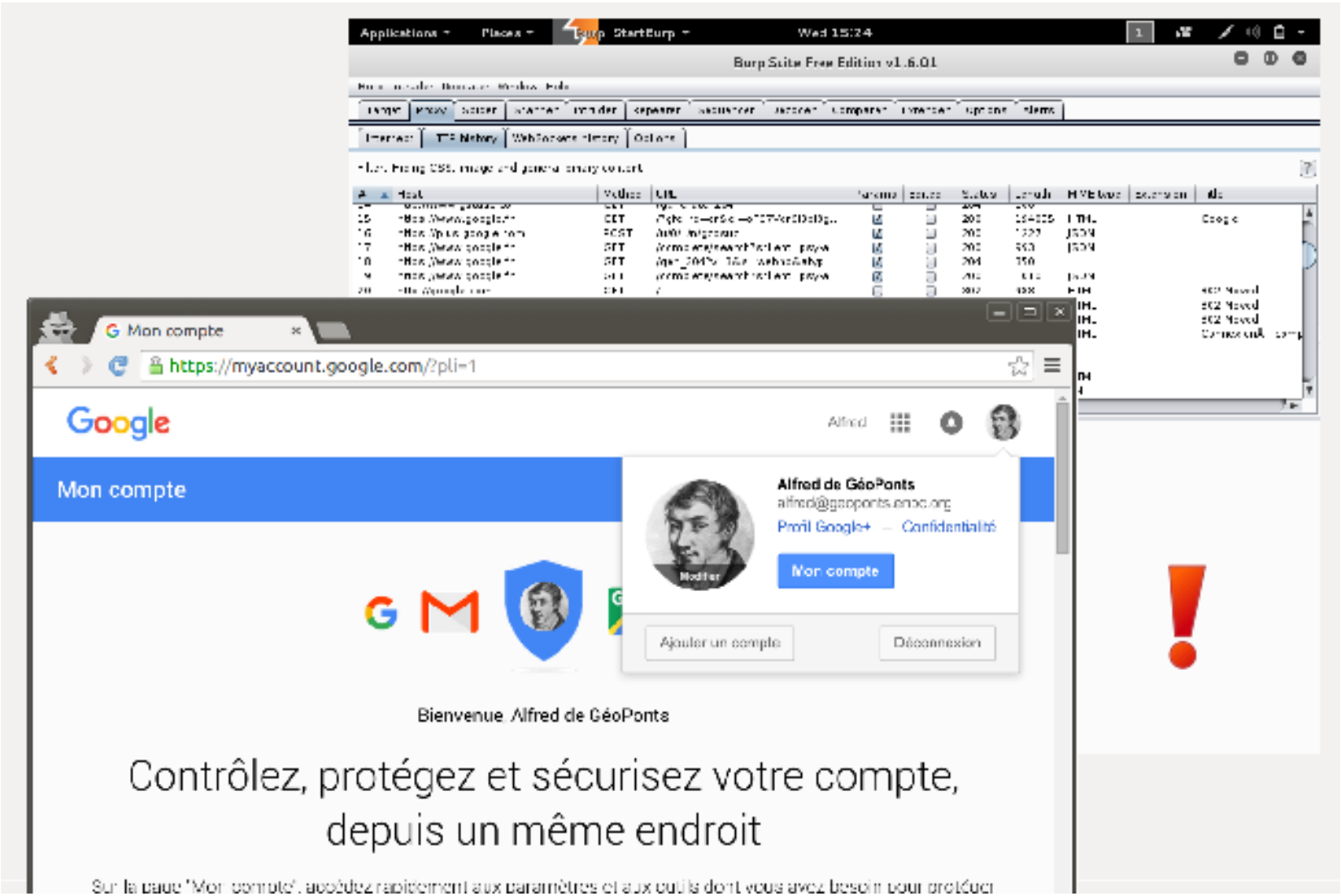
Connexion avec mot de passe à Google



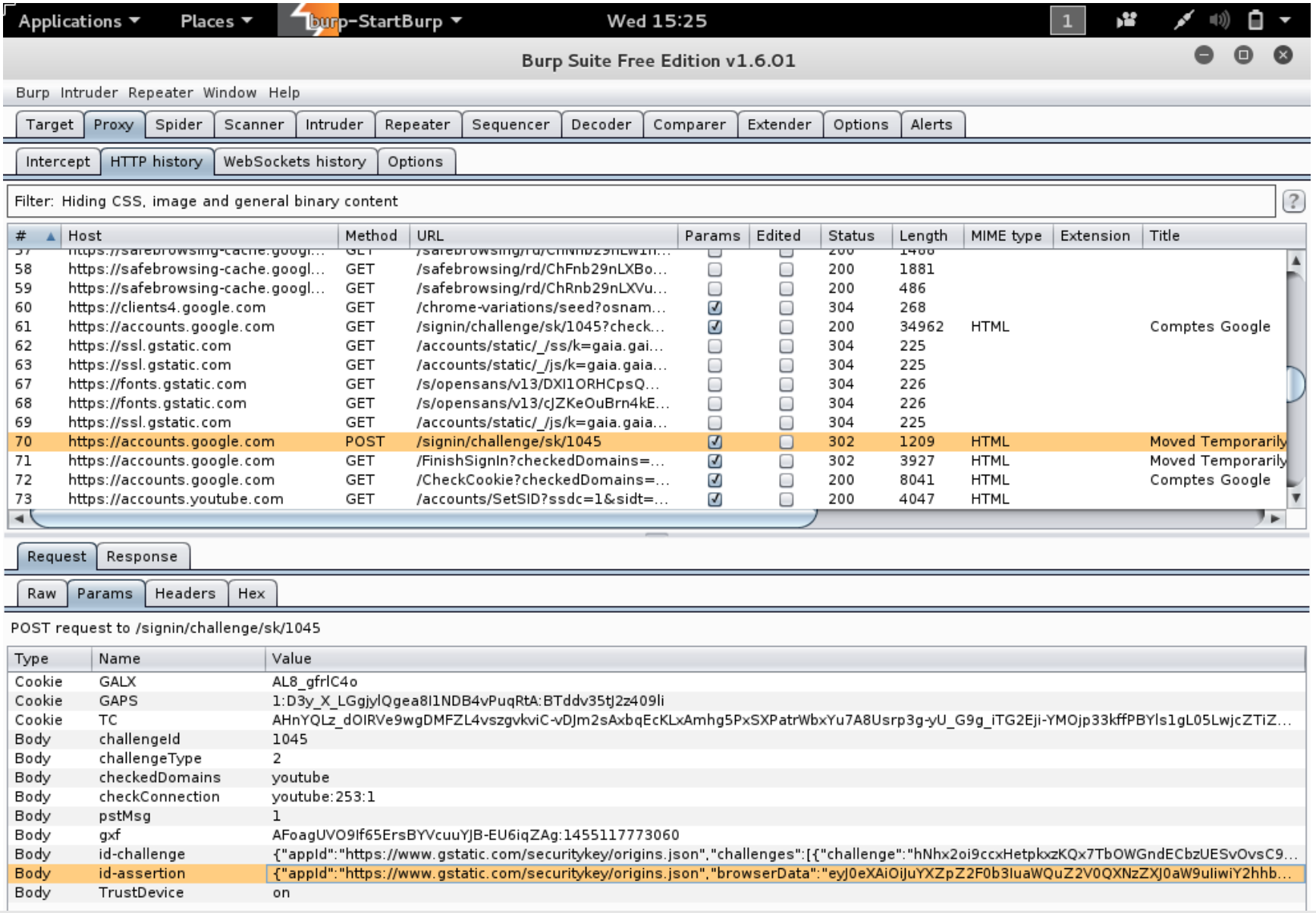
Authentification avec le token U2F



Authentification acceptée ! ? == MITM réussi !



Accès aux messages U2F transmis dans TLS



Signalement et qualification du risque résiduel

Contact avec Google :

- la vérification de TLS Channel ID est désactivée
  - « support expérimental par Chrome »
  - « usage incompatible avec les proxy-cache interceptants »
- vérification de l'efficacité effectuée avec un compte de test

Résultat net :

- TLS Channel ID n'est pas implémenté ou vérifié

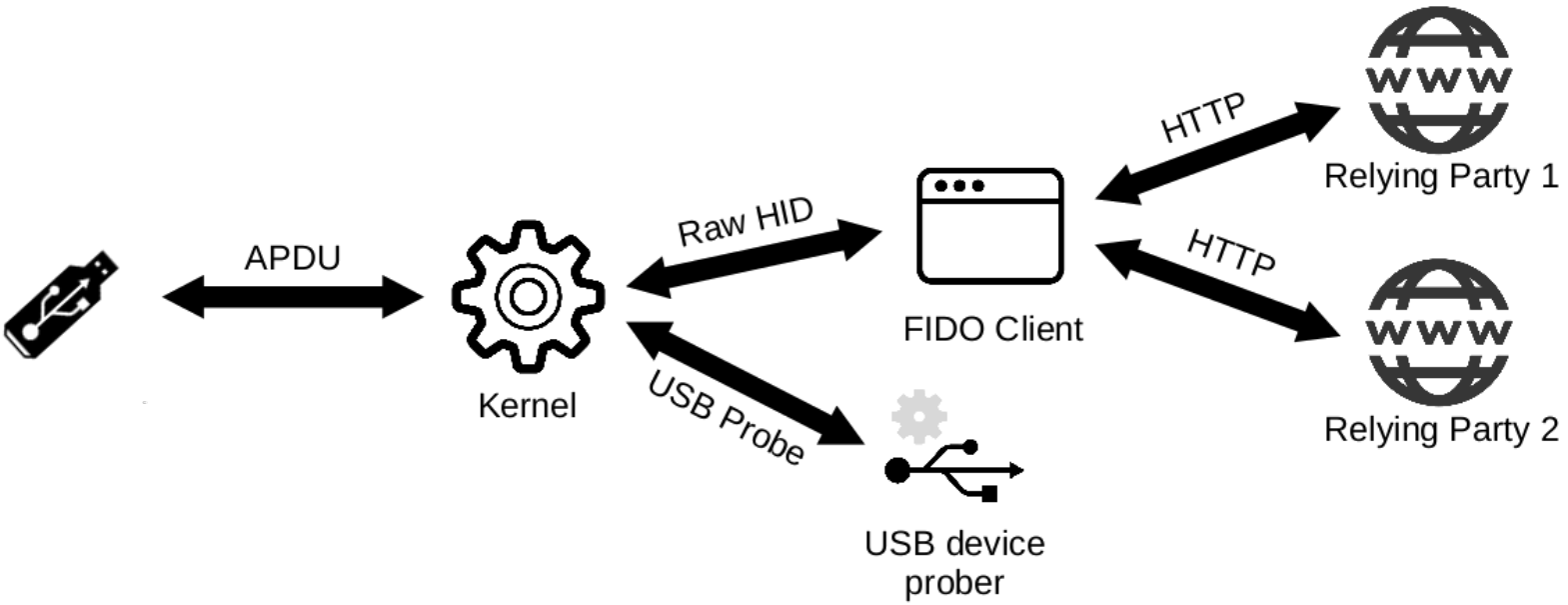
Autre limitation d'U2F :

le cas de la confirmation de transaction

Limite d'U2F assumée par les spécifications

Pas de confirmation de transaction

- pas de propriété WYSIWYS*



* : WYSIWYS : *What You See Is What You Sign*

Limite d'U2F assumée par les spécifications

Pas de confirmation de transaction

- pas de propriété WYSIWYS*

Contrairement à SMS et TOTP, pas de saisie utilisateur dans un champ affiché à l'écran :

- incertitude de ce que l'on confirmerait

Usage impossible, par exemple, pour valider une transaction bancaire (3DSecure)

* : WYSIWYS : *What You See Is What You Sign*

Comparatif avec

l'authentification TLS du client

par certificat

Facilité de mise en œuvre

Parmi les objectifs premiers d'U2F :

- être *plug 'n play*, sans pilote (RawHID)

Authentification par certificat :

- navigateurs coupables d'interfaces utilisateur kafkaesques
- utilisation simplifiable avec la complicité d'un service informatique

Résistance aux MITM TLS

U2F :

- en théorie, résistant grâce à TLS Channel ID*
- en pratique, pas de déploiement

Authentification par certificat :

- résistant*

* : L'attaque SLOTH (et autres attaques par collision de transcript) peut affecter U2F et l'authentification par certificat client

Protection contre des vulnérabilités web

U2F :

- utilisation de formulaires web
- interaction avec l'application sans authentification
- vol de session potentiel par XSS

Authentification par certificat :

- sans certificat valide, impossible de voir/communiquer avec l'application

Recul sur les protocoles

U2F :

- jeune (peu d'implémentations, peu de déploiements)
- cette étude est la première qui soit indépendante

Authentification par certificat :

- spécifié en 1995
- 3SHAKE, SLOTH, MS09-007...

Conclusions sur le protocole U2F

Conclusion

U2F :

- second facteur semblant satisfaisant pour le grand public
- facilité d'utilisation
- pas de confirmation de transaction : n'est pas un remplacement universel du SMS/TOTP
- nécessite des études supplémentaires :
  - pas d'étude des implémentations
  - pas d'étude de la fonctionnalité *facets*

Authentification par certificat :

- pas encore remplacé pour les applications sensibles

Q & A

Merci pour votre attention

Florian Maury
ANSSI

Mickaël Bergem
ParisTech