Identité et Contrôle d'Accès

Pascal.Urien@Telecom-ParisTech.fr

Identité et Authentification

"On the Internet, nobody knows you're a dog."

Identité et Authentification

- L'identité détermine les autorisations et/ou la localisation d'une personne, d'un objet (parfois intelligent), ou des deux, relativement à une organisation (état, privée…).
- L'authentification est la procédure qui consiste à faire la preuve d'une identité. On peut utiliser divers moyens,
  - Ce qui je suis, méthodes biométriques, éventuellement multi modales (passeport électronique…)
  - Ce que je connais, mots de passe, ….
  - Ce que je possède, cartes à puce, jeton…

Quelques exemples d'organisations gérant des d'identités

- **Les états** (identité des citoyens et des visiteurs). Les identités utilisent par exemple la biométrie (programme *US-Visit*), les passeports électroniques, les cartes d'identité munies de puces sécurisées.
- **Le WEB** (Applications WEB). Il existe de multiples infrastructures d'identités telles que, *Single Sign On* (SSO), Microsoft Passport, Liberty Alliance, OPENID…
- **L'industrie** (localisation, inventaire…). L'identité des objets est associée à des étiquettes (code barre, code 2D) ou des RFIDs.
- **Les services informatiques** gèrent des ordinateurs personnels. L'identité d'une machine est par exemple assurée par un *Trusted Platform Module (TPM)*.
- **Les réseaux bancaires** réalisent des opérations de paiement associées à des cartes EMV.
- **Les réseaux de communication** réalisent des échanges de données, et mettent en œuvre différentes identités : couple login mot de passe, carte SIM ou USIM , Wi-Fi et EAP-ID, jetons divers (tels que RSA SecurID)

Authentification

- Trois catégories,
  - Symétrique (secret partagé),
  - Asymétrique (RSA ou ECC).
  - Tunnel.
- Authentification Simple
  - Une entité de communication est authentifiée
- Authentification mutuelle
  - Les deux entités de communication sont mutuellement authentifiées

Mécanisme Symétrique

- **Les mots de passe**.
- Un mot de passe est une suite de caractères qui peut être facilement mémorisée par un être humain.
- Il ne s'agit pas d'un nombre aléatoire mais au contraire d'une concaténation de mots, de chiffres et de signes. De tels secrets peuvent être devinés à l'aide d'une attaque par dictionnaire, c'est à dire un programme utilisant une base de données pour la génération de ces valeurs.
- Ainsi la méthodes EAP-MSCHAPv2 repose sur la clé NT, c'est à dire l'empreinte MD4 (soit 16 octets) d'un mot de passe.

MOT de passe

MOST POPULAR PASSWORDS

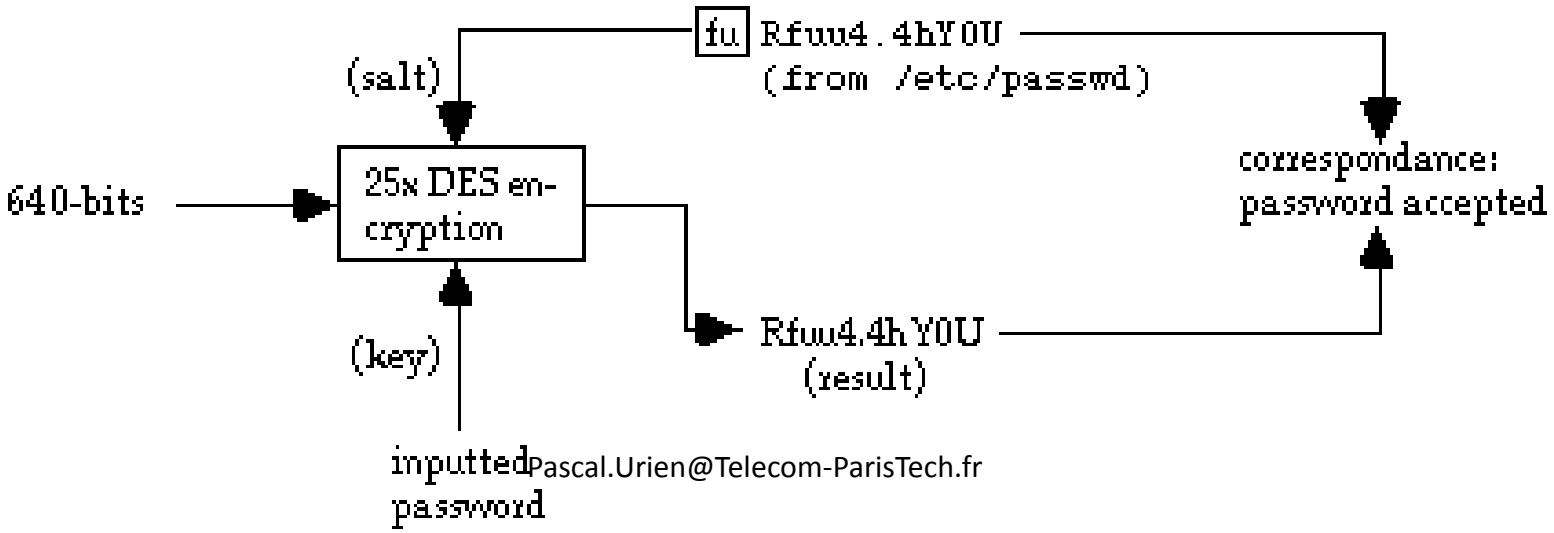
Nearly one million RockYou users chose these passwords to protect their accounts.

1. **123456**	17. **michael**
2. **12345**	18. **ashley**
3. **123456789**	19. **654321**
4. **password**	20. **qwerty**
5. **iloveyou**	21. **iloveu**
6. **princess**	22. **michelle**
7. **rockyou**	23. **111111**
8. **1234567**	24. **0**
9. **12345678**	25. **tigger**
10. **abc123**	26. **password1**
11. **nicole**	27. **sunshine**
12. **daniel**	28. **chocolate**
13. **babygirl**	29. **anthony**
14. **monkey**	30. **angel**
15. **jessica**	31. **FRIENDS**
16. **lovely**	32. **soccer**

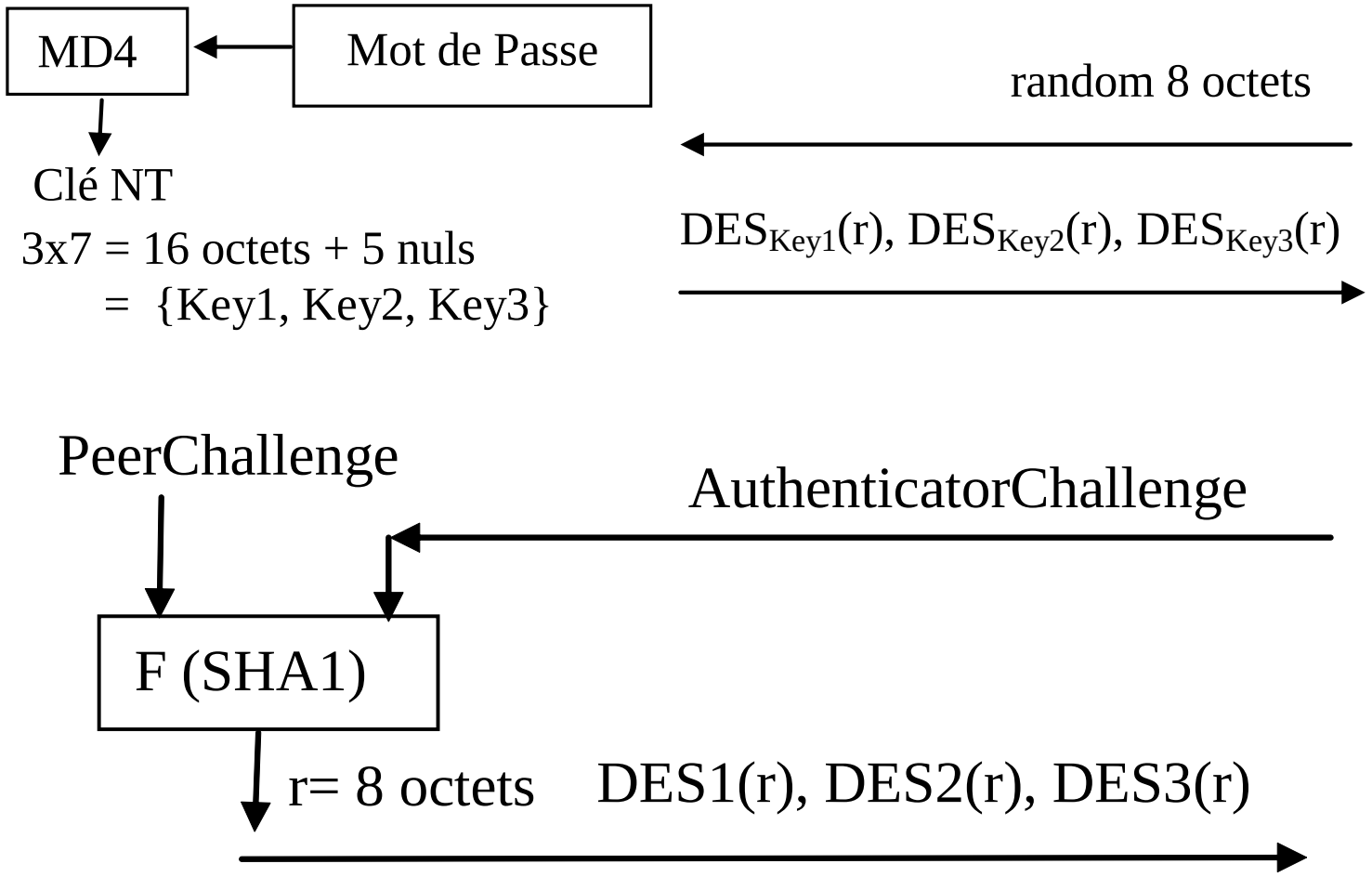
Source : Imperva

Unix

- Dans les anciens systèmes UNIX la liste des empreintes des mots de passe est stockée dans le ficher /etc/passwd.
- Le fichier /etc/shadow lisible uniquement en mode *root* est aujourd'hui préféré.
- Un mot de passe est par exemple une chaîne d'au plus 8 caractères ASCII, convertie en une clé DES de 56 bits (8 x 7bits).
- La fonction crypt(key, salt) réalise dans ce cas 25 chiffrements DES consécutifs (avec une valeur initiale IV=0), dont chacun comporte une permutation choisie parmi 4096 possible (soit 12 bits, c'est le paramètre *salt*, deux caractères choisis parmi 64 valeurs possibles *a-z A-Z 0-9 . /)*.
- D'autres algorithmes de génération d'empreinte sont supportés, tels que la fonction MD5.



Microsoft: MCHAPv1, MSCHAPv2



Comment réaliser une authentification mutuelle basée sur un mot de passe ?

Secret Partagé

- **Les secrets partagés** (PSK, Pre-Shared-Key).
- Il s'agit en fait d'un nombre aléatoire, dont la taille est l'ordre de 128 à 160 bits.
- Un cerveau humain éprouve beaucoup de difficultés à mémoriser ces informations.
- Le stockage sécurisé du secret est réalisé par exemple par le système d'exploitation d'un ordinateur personnel ou par une carte à puce.

Provisionning

- **Le mode provisioning**.
- Dans les réseaux GSM ou UMTS une base de donnée centralisée (*Host Location Register*) gère les comptes utilisateurs, en particulier leur PSK.
- Afin d'éviter des interrogations fréquentes les méthodes d'authentification (A3/A8, Milenage) sont conçues de telle manière que le site central puisse produire des vecteurs d'authentification (triplets GSM ou quadruplets UMTS), réutilisables par des agents de confiance (tels que les *Visitor Location Register* par exemple).

Mécanismes Asymétriques

- Ces procédures sont basées sur des algorithmes tels que ECC, RSA ou Diffie-Hellman.
- Le protocole SSL/TLS est généralement utilisé pour le transport de ces échanges.
- ! DH 512 bits (Z/pZ) cassé en 2015
  - Logjam attaque

1 < k < 2 N= modulo = 2^n

$$\exp\left((\tilde{k} + o(1))(\log N)^{1/3}(\log\log N)^{2/3}\right)$$

Logjam Attaque

- To carry out this attack, we implement the number field sieve discrete log algorithm.
- After a week-long precomputation for a specified 512-bit group, we can compute arbitrary discrete logs in that group in about a minute.
- We find that 82% of vulnerable servers use a single 512-bit group, allowing us to compromise connections to 7% of Alexa Top Million HTTPS sites.
- In response, major browsers are being changed to reject short groups. We go on to consider Diffie-Hellman with 768- and 1024-bit groups.
- We estimate that even in the 1024-bit case, the computations are plausible given nation-state resources.
- A small number of fixed or standardized groups are used by millions of servers; performing precomputation for a single 1024-bit group would allow passive eavesdropping on 18% of popular HTTPS sites, and a second group would allow decryption of traffic to 66% of IPsec VPNs and 26% of SSH servers.
- A close reading of published NSA leaks shows that the agency's attacks on VPNs are consistent with having achieved such a break.
- We conclude that moving to stronger key exchange methods should be a priority for the Internet community.

Les tunnels

- Ainsi que nous l'avons souligné précédemment les méthodes d'authentification basées sur des mots de passe, sont sujettes à des attaques par dictionnaire.
- Ainsi le protocole MSCHAP assure une protection jugée raisonnable dans des environnements sures (par exemple des intranets ou des connexions par modem), mais n'est plus adapté lors d'une mise en œuvre dans un milieu hostile, tel que IP sans fil, abritant potentiellement de nombreuses oreilles électroniques indiscrètes.
- Les tunnels, s'appuyant fréquemment sur la technologie SSL/TLS, protègent un dialogue d'authentification grâce au chiffrement des données échangées.
- Il existe aujourd'hui de multiples standards devenus nécessaires, en raison des nombreux logiciels disponibles sur le WEB qui cassent les protocoles à base de mot de passe.

HTTP: Pourquoi le login/mot de passe ?

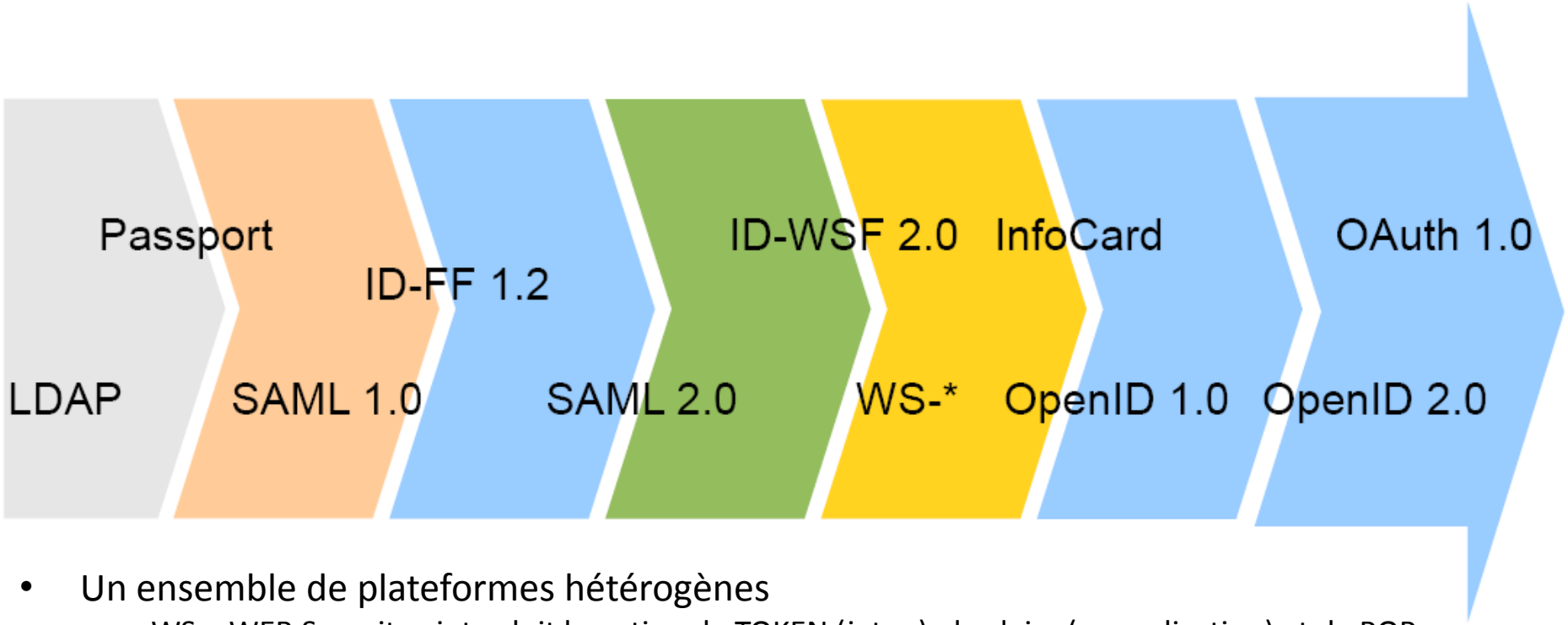
- Un héritage de l'informatique traditionnelle pour la gestion des comptes utilisateurs
- Le mot de passe est une clé symétrique partagée avec le serveur
  - **Il devrait y avoir autant de secrets que de serveurs, mais c'est difficile à mettre en œuvre pour un utilisateur humain**.
- NETSCAPE a inventé le protocole SSL en 1995, pour sécuriser les sessions HTTP
  - Le serveur est identifié par un certificat
  - Le client est identifié par le couple login/mot de passe
  - La session SSL sécurise le mot de passe
- Le cookie a été inventé en 1995 avec la version 2 de Netscape
  - Le cookie réalise l'authentification/autorisation d'une session HTTP, c'est-à-dire d'une suite de requêtes, il est inséré dans l'en tête HTTP
- Le Session ID (sid) joue également le rôle du cookie, il peut être inséré dans l'entête d'une requête GET HTTP ou dans le corps du message pour une méthode POST
  - En PHP les cookies et les sid (SessionID) jouent le même rôle d'identifiant de session

HTTP: Comment sécuriser un mot de passe

- HTTP digest, RFC 2617
  - KD(secret, data) = H(concat(secret, ":", data))
  - Déployé, mais pas utilisé en pratique
- RFC 4279, TLS-PSK (pre shared key)
  - PreMasterSecret = nonce | PSK
  - Le nonce est transmis chiffré avec la clé publique du serveur
  - Pas de déploiement connu
- Le One Time Password (OTP)
  - hash(timestamp | secret), par exemple RSA SecureID
  - MAC(compteur, secret), RFC 4226 , "HOTP: An HMAC-Based One-Time Password Algorithm"
  - MAIS sensible aux attaques MIM (*Man In the Middle*), autrement dit vol d'un OTP.
- Solution possible
  - Utiliser un OTP comme le PSK d'une session TLS-TSK. Le serveur estime les valeurs possibles de l'OTP.

Systèmes de Gestion d'identités

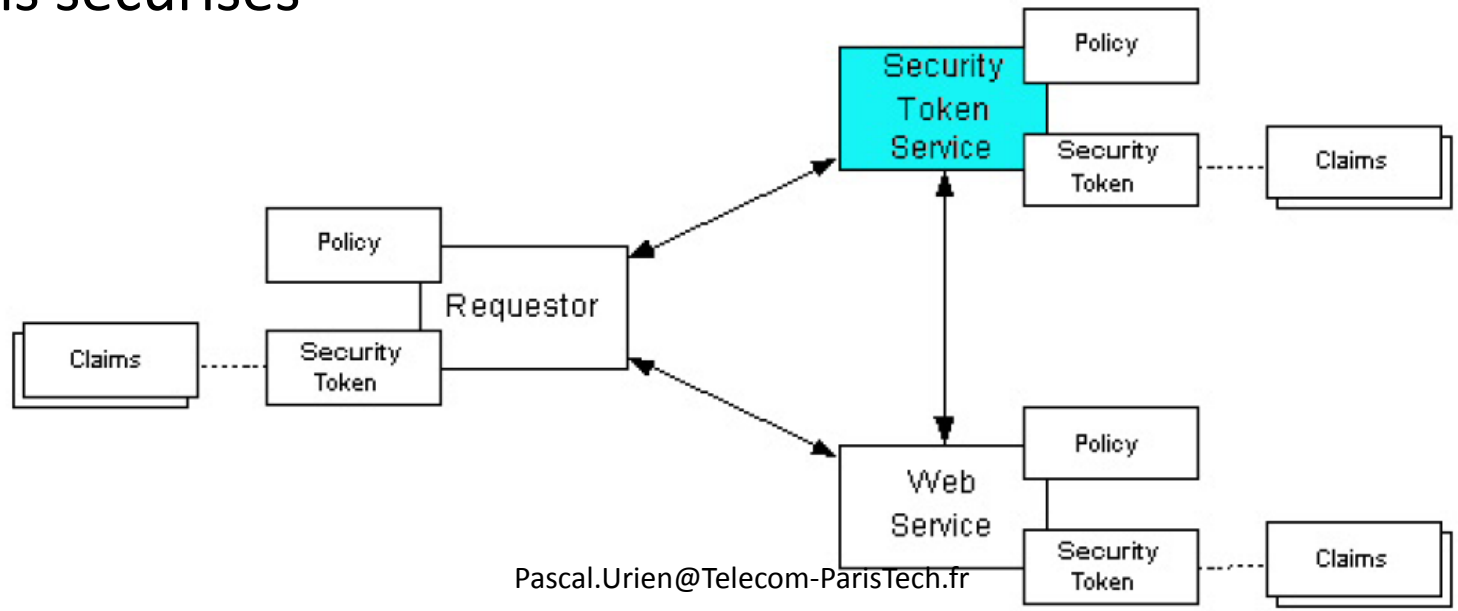
Technologies de Gestion d'Identités



- Un ensemble de plateformes hétérogènes
  - WS_, WEB Security : introduit la notion de TOKEN (jeton), de claim (revendication) et de POP (*Proof-of-Possession Token)*
  - Liberty Alliance, fédération d'identité, définition du concept d'*Identity Provider (*IdP). Adoption du standard SAML.
  - SAML, *Security Assertion Markup Language*, introduit la notion d'assertion SAML (un jeton réalisé en syntaxe XML), et de transport (*Protocol Binding*)
  - InfoCard(Microsoft), un *Information Card* est un artefact qui contient des métadonnées sous la forme d'un jeton prouvant la relation entre un *Identity Provider* et un utilisateur
  - OPENID, un standard de SSO (*Single Sign On*), utilisant implicitement la notion de jeton sécurisé
- Dans la pratique, le mot de passe reste roi pour l'authentification

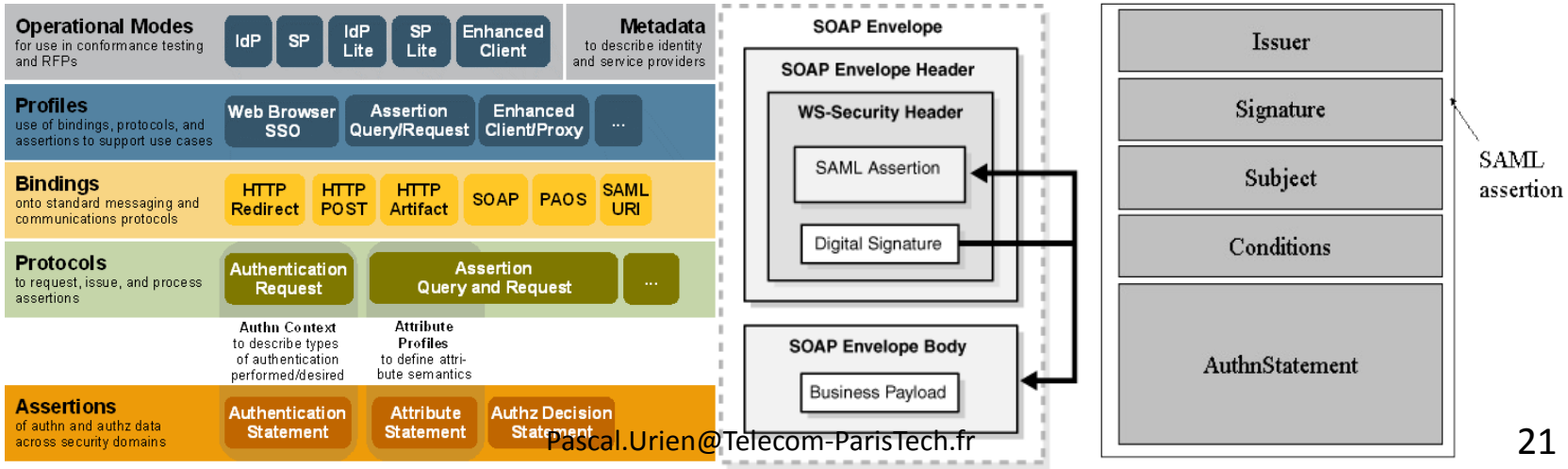
WS_TRUST: Web Services Trust Langage

- *Claim* (droit): une demande relative à un client, un service ou tout autre ressource (nom, identité, clés, privilèges…)
- *Security Token*: un ensemble de claims
- *Signed Security Token*, un jeton sécurisé muni d'une signature
- *Proof-of-Possession Token*: la preuve (POP) de possession d'un jeton sécurisé, typiquement à l'aide d'une clé cryptographique
- *Security Token Service* (STS): un service WEB qui délivre des jetons sécurisés



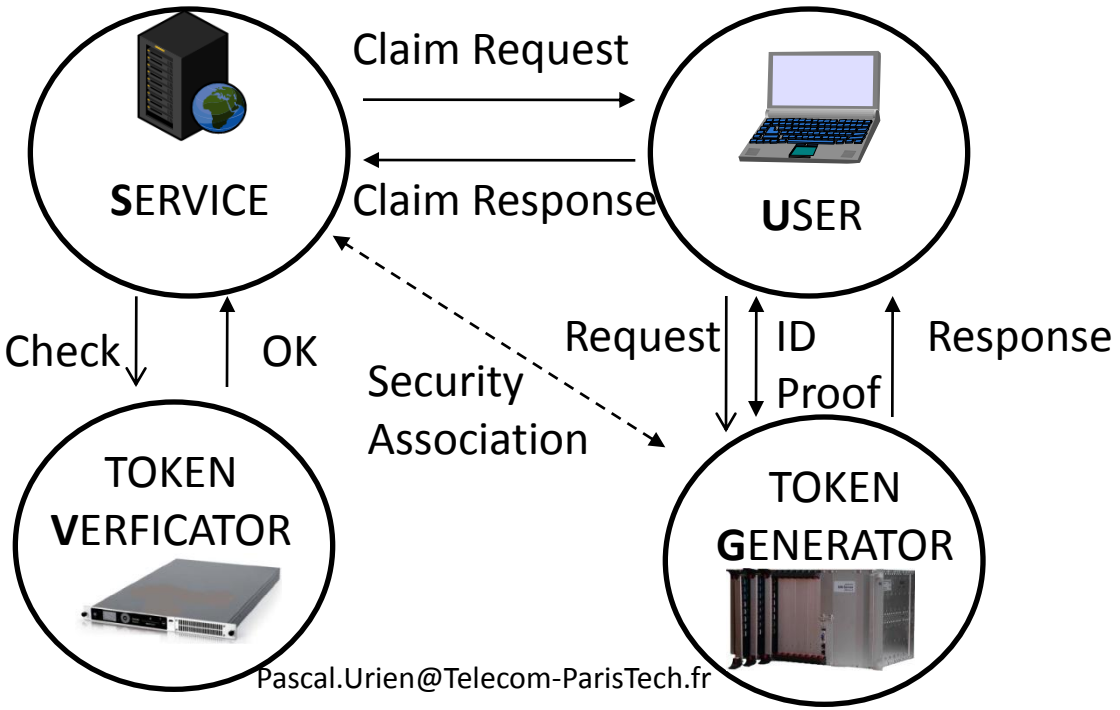
Security Assertion Markup Language

- Une syntaxe basée sur XML, dédié à l'authentification de l'utilisateur d'un service et des attributs associés.
- SAML est utilisé par les projets Liberty Alliance, Shibboleth (Internet 2) et WS_Security (OASIS Web Services Security)
- SAML introduit les notions d'assertion, de « protocols binding », et de profile
  - Une assertion est un ensemble de données généré par une autorité SAML
    - Authentification, une preuve d'identité délivrée par un Identity Provider
    - Attributs, un ensemble d'attribut liés à un sujet
    - Décision d'autorisation, un droit pour accéder à une ressource
  - Un ensemble de protocoles de type requêtes/réponse réalisant, entre autres, authentification et délivrance d'assertions
  - Bindings, transport des messages SAML par des protocoles tels que SOAP
  - Profiles, définition de l'usage de SAML pour des applications particulières (Web SSO, etc…)

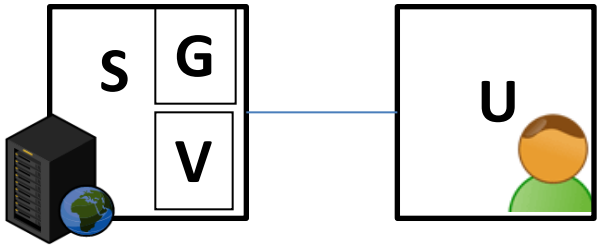


Le Token: un cookie d'Autorisation

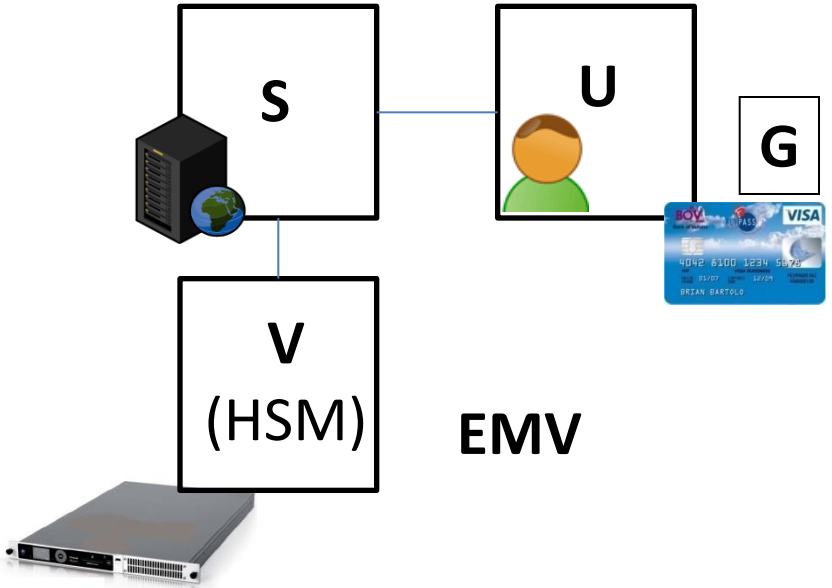
- Un Token (jeton) est une liste de revendications (claims)
  - Un ensemble d'information sécurisé, exprimé à l 'aide d'une syntaxe *XML* ou *URLEncoded*
- Délivrance de Tokens
  - Nécessite une preuve d'identité de l'internaute
  - Mot de passe, certificat, …
- Vérification de Tokens



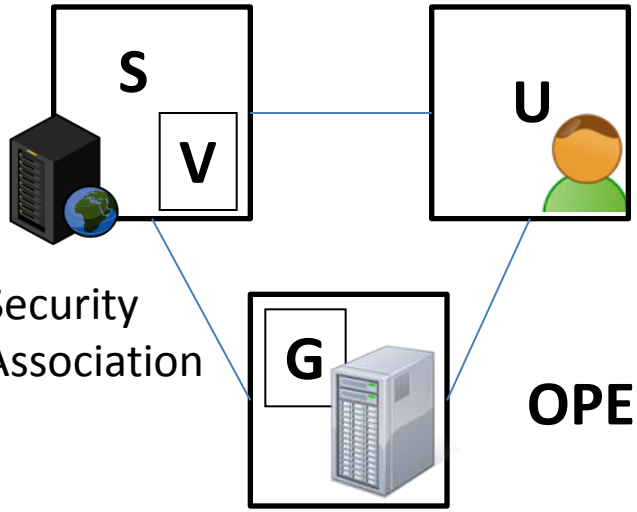
Quelques Architectures



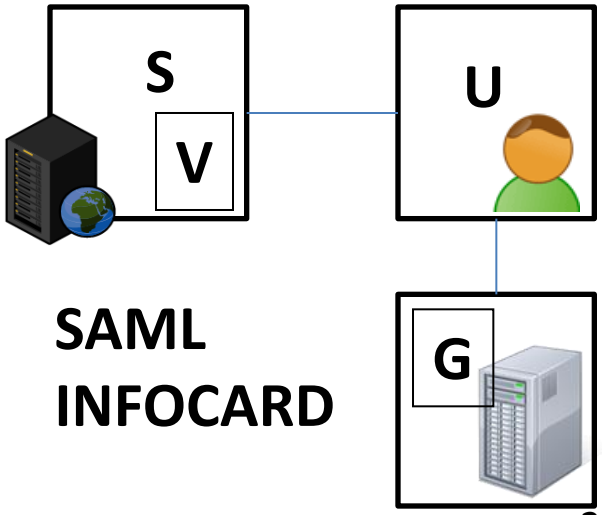
Login / Password



EMV

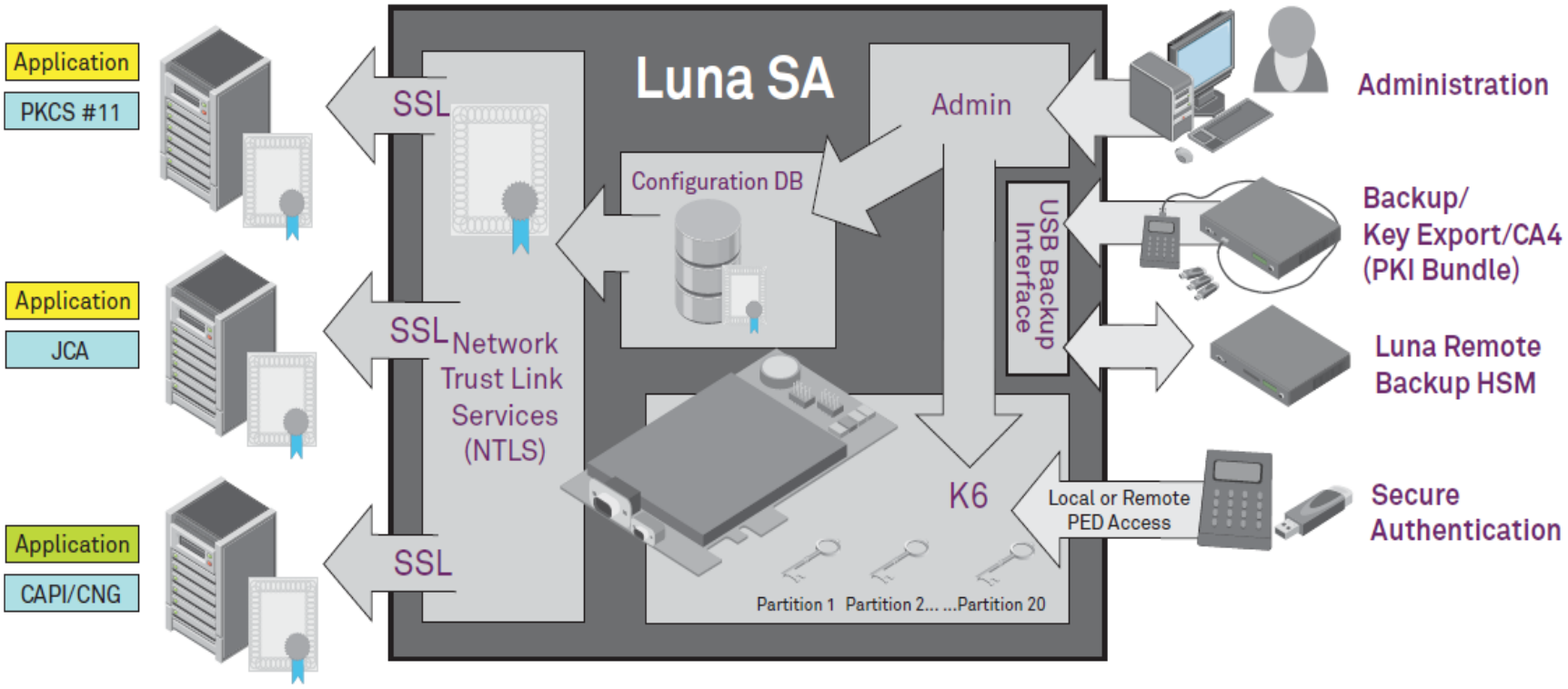


OPENID



SAML
INFOCARD

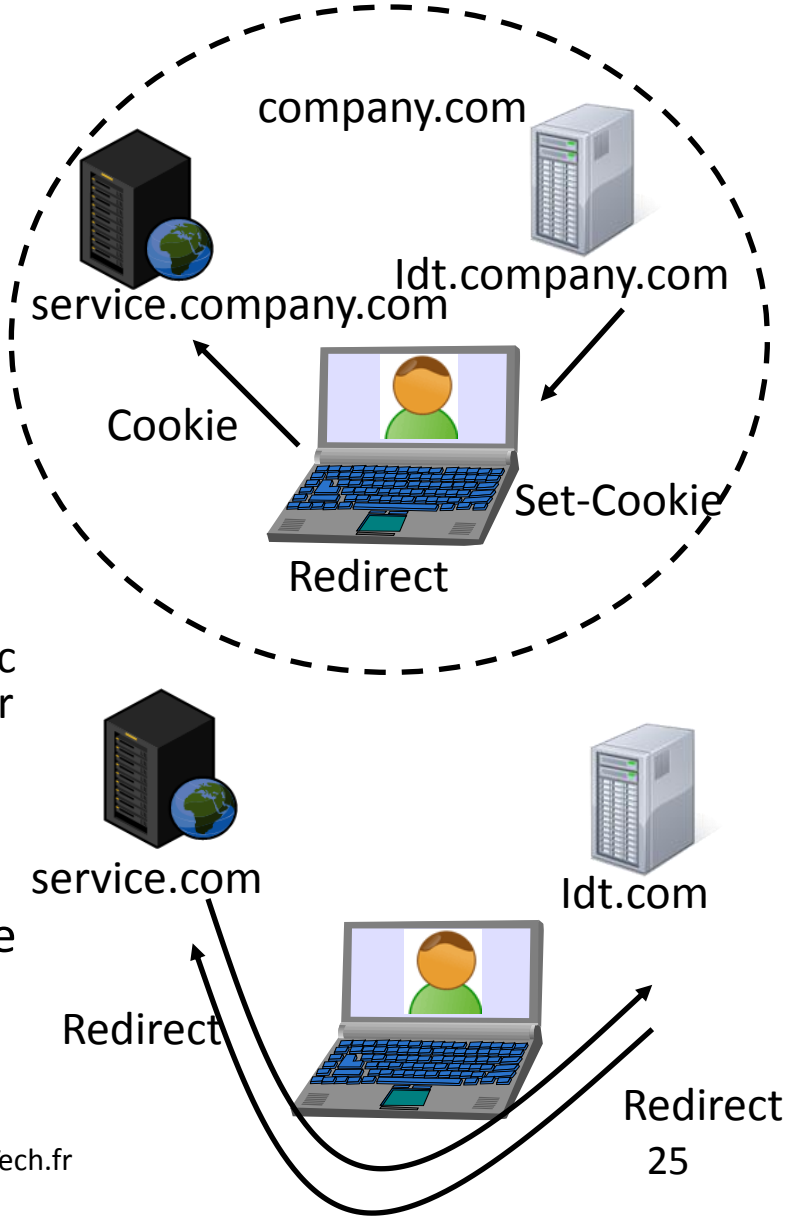
Hardware Secure Module



J2EE Connector Architecture (JCA) Cryptography API: Next Generation (CNG)

Comment Transférer un Token

- Problème
  - Le service et le générateur de jetons ne sont pas toujours dans le même domaine
- Infrastructure privé
  - Service.company.com et Idt.company.com sont deux sous domaines de company.com
  - Il est possible d'échanger un cookie entre sous domaines
- Infrastructure publique
  - Les cookies inter-domaines sont interdits
  - Une mécanisme couramment utilisé est la redirection HTTP (voir par exemple *OPENID*) avec un point de pivot via le navigateur de l'utilisateur (*USER*)
    - Cette approche engendre un problème de privacy en raison du lien entre le service et le serveur d'authentification
- Le navigateur du client est le point central d'une architecture de gestion d'identité dans un contexte *Cloud Computing*
  - En particulier il assure le stockage des Tokens

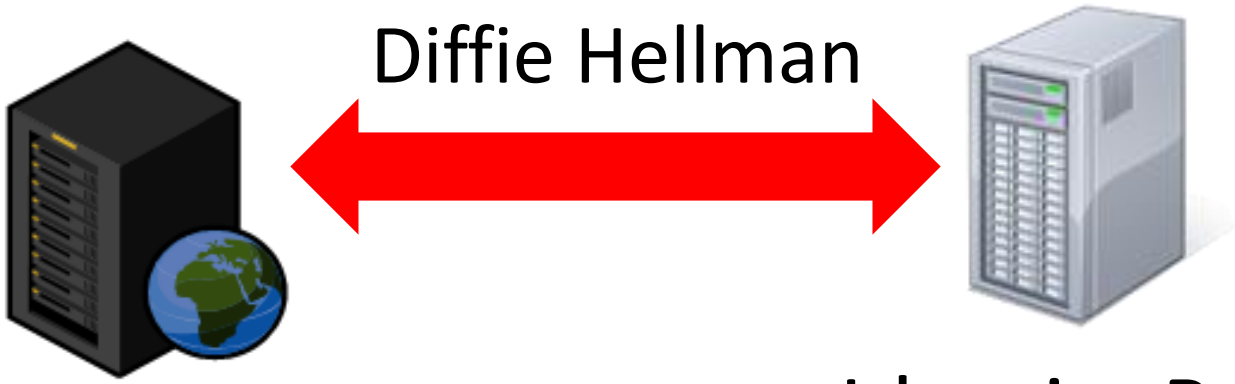


Architectures de Single Sign On

- De nombreuses organisations proposent des applications accessibles à travers des interfaces WEB
  - Les identités des utilisateurs (login, password, certificats) sont stockées sur des bases de données.
  - Le processus d'authentification consiste à vérifier l'identité d'un utilisateur afin d'établir ses droits relativement à un service disponible.
  - On désigne par "Sign On" l'opération d'identification / authentification d'un utilisateur du service.
  - Une architecture "Single Sign On" permet d'éviter de multiples d'opérations " Sign On".
- Exemples
  - .netPassport, Microsoft
    - Une identité unique pour tous les services
  - Liberty Alliance
    - Une fédération d'identités pour de multiples services.
  - OPENID

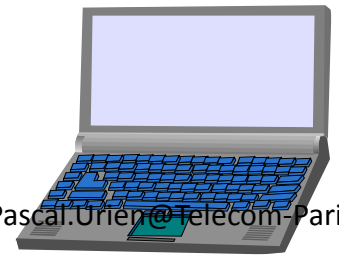
OPENID

- Single Sign ON
- 1 million de sites WEB compatible



Consumer

Identity Provider



User

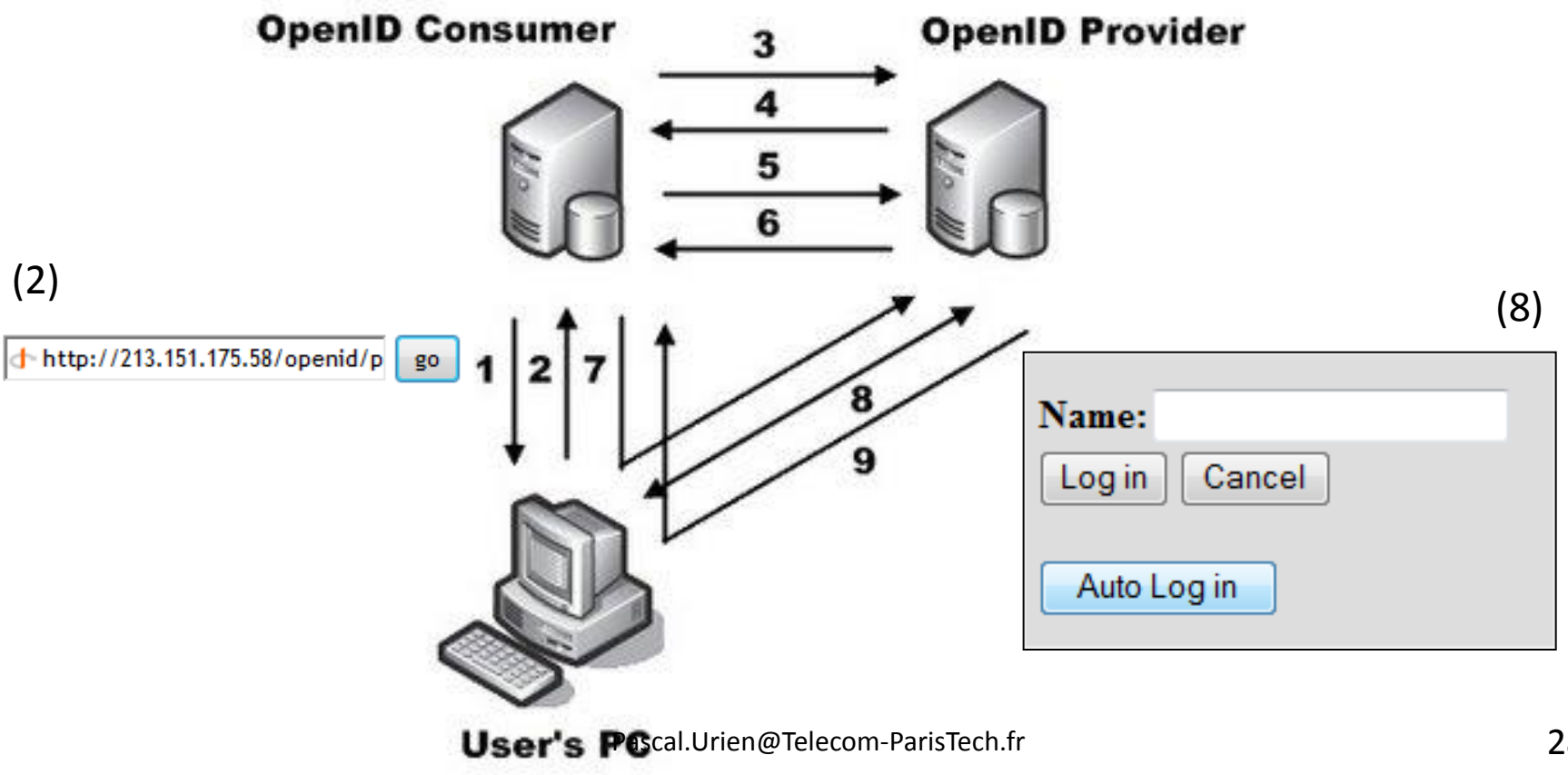
OPENID

(6) Association de Sécurité

The association handle

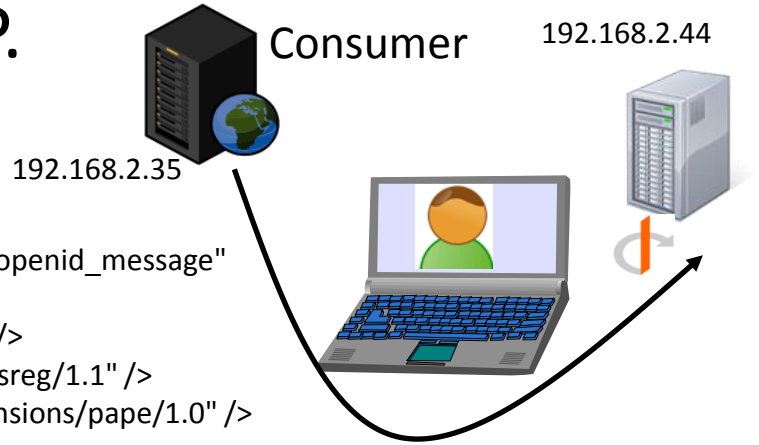
The mac encryption key,

• Encrypted key = sha1(g^{xy} mod p) exor mac-key



Exemple de *Claim Request*, OpenID

- Le service (consumer) délivre un *Claim Request* à l'intention de l'Identity Provider, à l'aide d'un mécanisme de redirection HTTP.

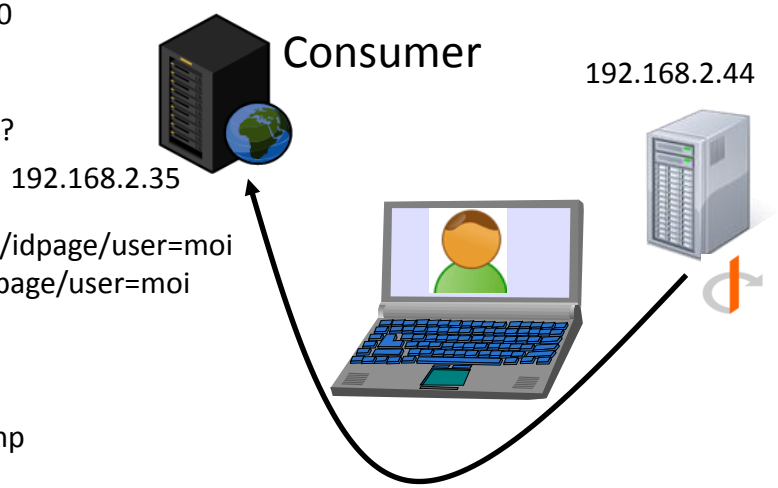



```
<html><head><title>OpenId transaction in progress</title></head>
<body onload='document.forms[0].submit();'>
<form accept-charset="UTF-8" enctype="application/x-www-form-urlencoded" id="openid_message"
action="http://192.168.2.44/openid/examples/server/server.php" method="post">
<input type="hidden" name="openid.ns" value="http://specs.openid.net/auth/2.0" />
<input type="hidden" name="openid.ns.sreg" value="http://openid.net/extensions/sreg/1.1" />
<input type="hidden" name="openid.ns.pape" value="http://specs.openid.net/extensions/pape/1.0" />
<input type="hidden" name="openid.sreg.required" value="nickname" />
<input type="hidden" name="openid.sreg.optional" value="fullname,email" />
<input type="hidden" name="openid.pape.preferred_auth_policies" value="" />
<input type="hidden" name="openid.realm" value="http://192.168.2.35:80/openid/examples/consumer/" />
<input type="hidden" name="openid.mode" value="checkid_setup" />
<input type="hidden" name="openid.return_to"
value="http://192.168.2.35:80/openid/examples/consumer/finish_auth.php?janrain_nonce=2009-03-17T09:07:14ZGoN7wY" />
<input type="hidden" name="openid.identity" value="http://192.168.2.44/openid/examples/server/server.php/idpage?user=moi" />
<input type="hidden" name="openid.claimed_id" value="http://192.168.2.44/openid/examples/server/server.php/idpage?user=moi" />
<input type="hidden" name="openid.assoc_handle" value="{HMAC-SHA1}{49bf5e64}{Va0OCQ==}" />
<input type="submit" value="Continue" />
</form>
<script>var elements = document.forms[0].elements;for (var i = 0; i < elements.length; i++) {  elements[i].style.display = "none";}
</script></body></html>
```

Exemple de *Claim Response*, OpenID

- Le *Identity Provider* délivre un *token* au service (*Consumer*), à l'aide d'un mécanisme de redirection HTTP.

HTTP/1.1 302 Found
Date: Tue, 17 Mar 2009 09:07:28 GMT
Server: Apache/2.2.6 (Win32) DAV/2 mod_ssl/2.2.6 OpenSSL/0.9.8g mod_autoindex_color PHP/5.2.5
X-Powered-By: PHP/5.2.5
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
location: http://192.168.2.35:80/openid/examples/consumer/finish_auth.php?
janrain_nonce=**2009-03-17T09:07:14ZGoN7wY**
&openid.assoc_handle={HMAC-SHA1}{**49bf5e64**}{**Va0OCQ==**}
&openid.claimed_id=http://192.168.2.44/openid/examples/server/server.php/idpage/user=moi
&openid.identity=http://192.168.2.44/openid/examples/server/server.php/idpage/user=moi
&openid.mode=id_res
&openid.ns=http://specs.openid.net/auth/2.0
&openid.ns.sreg=http://openid.net/extensions/sreg/1.1
&openid.op_endpoint=http://192.168.2.44/openid/examples/server/server.php
&openid.response_nonce=**2009-03-17T09:07:28Z2Z6LCL**
&openid.return_to=http://192.168.2.35:80/openid/examples/consumer/finish_auth.php/janrain_nonce=**2009-03-17T09:07:14ZGoN7wY**
&openid.sig=HUBr4K7Ff51FHCywZFux2IcZ9Xg=
&openid.signed=assoc_handle,claimed_id,identity,mode,ns,ns.sreg,op_endpoint,response_nonce,return_to,signed,sreg.email,sreg.fullname,
sreg.nickname
&openid.sreg.email=invalid@example.com
&openid.sreg.fullname=Example+User
&openid.sreg.nickname=example
Connection: close
Content-Length: 0
Content-Type: text/html



Microsoft .net Passport

- Un passeport est une collection de deux classes d'informations *Credential et Profile*, identifiée par un PUID (*Passport User ID*, 64 bits)
- Les messages d'authentification sont échangés sous forme de tickets, insérés dans des *cookies*.

Information	Data Type	Required to create a Passport?	Shared with other sites?
email address (Sign in name)	Credential and Profile	Yes	If user opts-in
Password	Credential	Yes	Never
Secret Questions and Answers	Credential	Optional	Never
Mobile Phone Number and Mobile PIN	Credential	Context dependant	Never
Security Key	Credential	Optional	Never
Birth Date, Country/Region, First Name, Gender, Last Name, Occupation, Postal Code, Preferred Language, State, Time Zone	Profile	Optional	If user opts-in

Microsoft .net Passport

Service Provider

Accords Contractuels
(SERVICE PROVIDER Key

Participant Cookie
{Ticket}ServiceProviderKey

{Ticket}Service Provider Key

4
Sign In

3
TGT

1
Passport
login

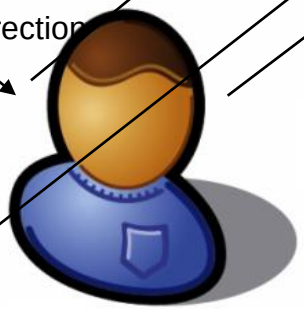
Site de gestion
des passeports

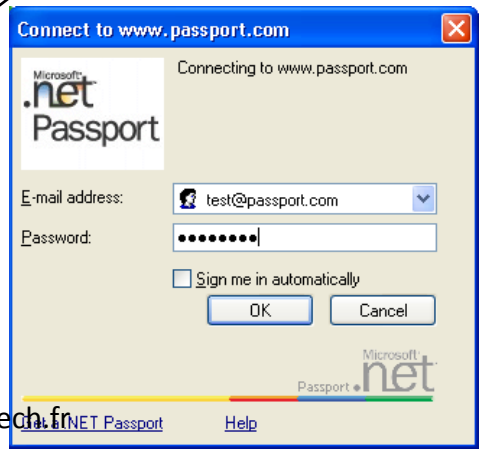
{PUID,Password}SSL

HTTP
Re-direction

Service Provider
Login

2





Cookie du Service Provider
{Ticket}Service Provider Key

Cookie d'Authentification
TGT, Ticket-Granting Cookie

Mais … des “bugs” 3/3

- Usurpation d’identité par vol de cookie (durée de validité d’un cookie de l’ordre de 15 mn)
- Failles dans la confidentialité des informations stockées dans le passeport découvertes par *Marc Slemko* (en 30 mn!) en Novembre 2001.
- Richard Purcell (Microsoft) le 01/10/2001 : « nous ne pouvons pas garantir une sécurité totale sur Passport ».
  - http://www.men.minefi.gouv.fr/webmen/revuedeweb/passport.html
- Mai 2003, Muhammad Faisal Rauf Danka, (étudiant pakistanais) accède à tous les comptes *passport*, en insérant la chaîne *emailpwdreset* dans une URL d’accès
  - https://register.passport.net/emailpwdreset.srf?lc=1033&m=victim@hotmail.com&id=&cb=&prefem=attacker@attacker.com&rst=1
  - L’attaquant (*attacker@attacker.com*) reçoit un courrier électronique contenant une URL qui lui permet de changer le mot de passe du compte cible.

Mais … des “bugs” 3/3

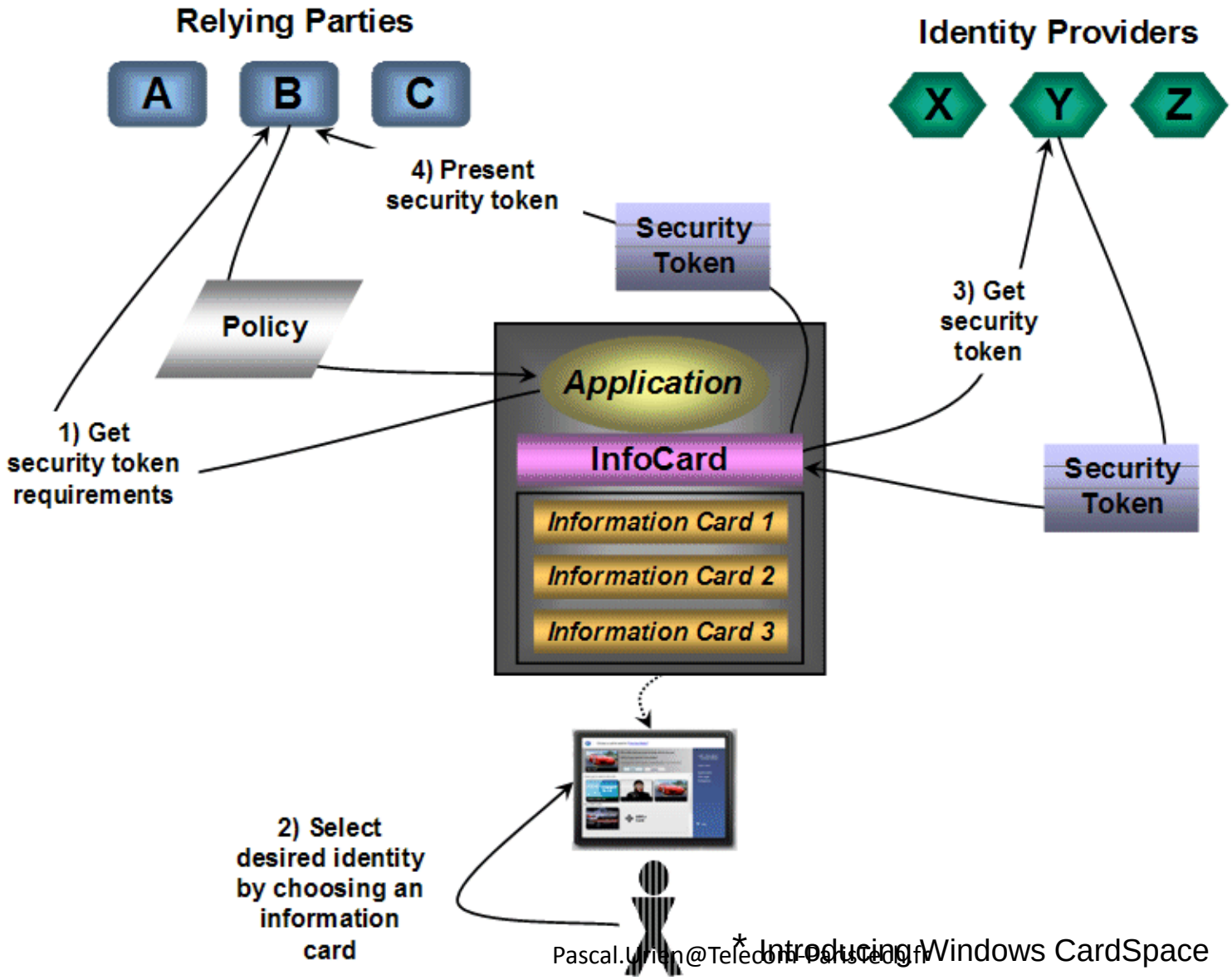
- Usurpation d’identité par vol de cookie (durée de validité d’un cookie de l’ordre de 15 mn)
- Failles dans la confidentialité des informations stockées dans le passeport découvertes par *Marc Slemko* (en 30 mn!) en Novembre 2001.
- Richard Purcell (Microsoft) le 01/10/2001 : « nous ne pouvons pas garantir une sécurité totale sur Passport ».
  - http://www.men.minefi.gouv.fr/webmen/revuedeweb/passport.html
- Mai 2003, Muhammad Faisal Rauf Danka, (étudiant pakistanais) accède à tous les comptes *passport*, en insérant la chaîne *emailpwdreset* dans une URL d’accès
  - https://register.passport.net/emailpwdreset.srf?lc=1033&m=victim@hotmail.com&id=&cb=&prefem=attacker@attacker.com&rst=1
  - L’attaquant (*attacker@attacker.com*) reçoit un courrier électronique contenant une URL qui lui permet de changer le mot de passe du compte cible.

Information Card (Microsoft)

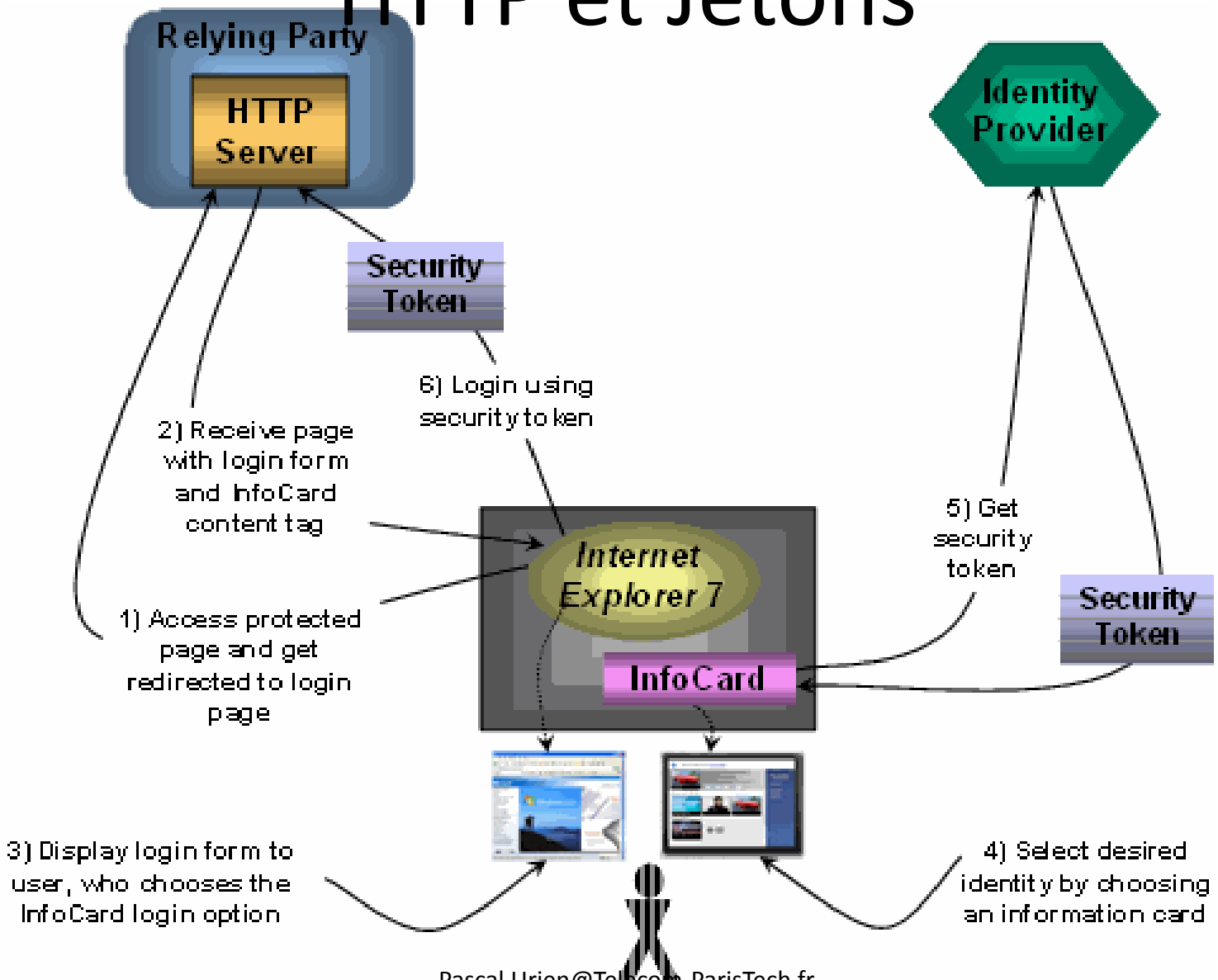
- An information card represents a digital identity of a user issued by an identity provider.
- Multiple digital identities for a user from the same identity provider are represented by different information cards.
- Users may obtain an information card from an identity provider, and may have a collection of information cards from various identity providers.
- An information card is simply an artifact that contains metadata and represents the token-issuance relationship between an identity provider and a user.
- It serves the very important purpose of transforming something abstract like digital identity into something concrete and tangible for users to work with in their digital interactions.
- Furthermore, being concrete entities, they are portable and can be carried around by a user to be used from any computer or device through which Web services are accessed.

*A Technical Reference for the Information Card Profile V1.0

Usage de jetons



HTTP et Jetons



* Introducing Windows CardSpace

Example de binding XML

<ic:InformationCard xml:lang="xs:language" ...>
<ic:InformationCardReference> … </ic:InformationCardReference>
<ic:CardName> xs:string </ic:CardName>?
<ic:CardImage MimeType="xs:string"> xs:base64Binary
</ic:CardImage>
<ic:Issuer> xs:anyURI </ic:Issuer>
<ic:TimeIssued> xs:dateTime </ic:TimeIssued>
<ic:TimeExpires> xs:dateTime </ic:TimeExpires>
<ic:TokenServiceList> ... </ic:TokenServiceList>
<ic:SupportedTokenTypeList> ... </ic:SupportedTokenTypeList>
<ic:SupportedClaimTypeList> ... </ic:SupportedClaimTypeList>
<ic:RequireAppliesTo ...> ... </ic:RequireAppliesTo> ?
<ic:PrivacyNotice ...> ... </ic:PrivacyNotice> ?
...
</ic:InformationCard>

1/3

- Objectifs
  - Fédération* de multiples identités associées à des fournisseurs de services WEB.
    - L'identité la plus commune est un couple (login, password). Le login peut être un identifiant de compte ou un NAI (Network Access Identifier, RFC 2486), similaire à une adresse de courrier électronique (la partie située à gauche de @ représente un identifiant de compte, la partie droite est le nom du domaine (serveur) gérant ce compte).
  - Une seule procédure d'authentification est suffisante pour accéder à un ensemble de services (Single Sign On).
  - L'infrastructure assure l'anonymat de l'utilisateur entre services fédérés.

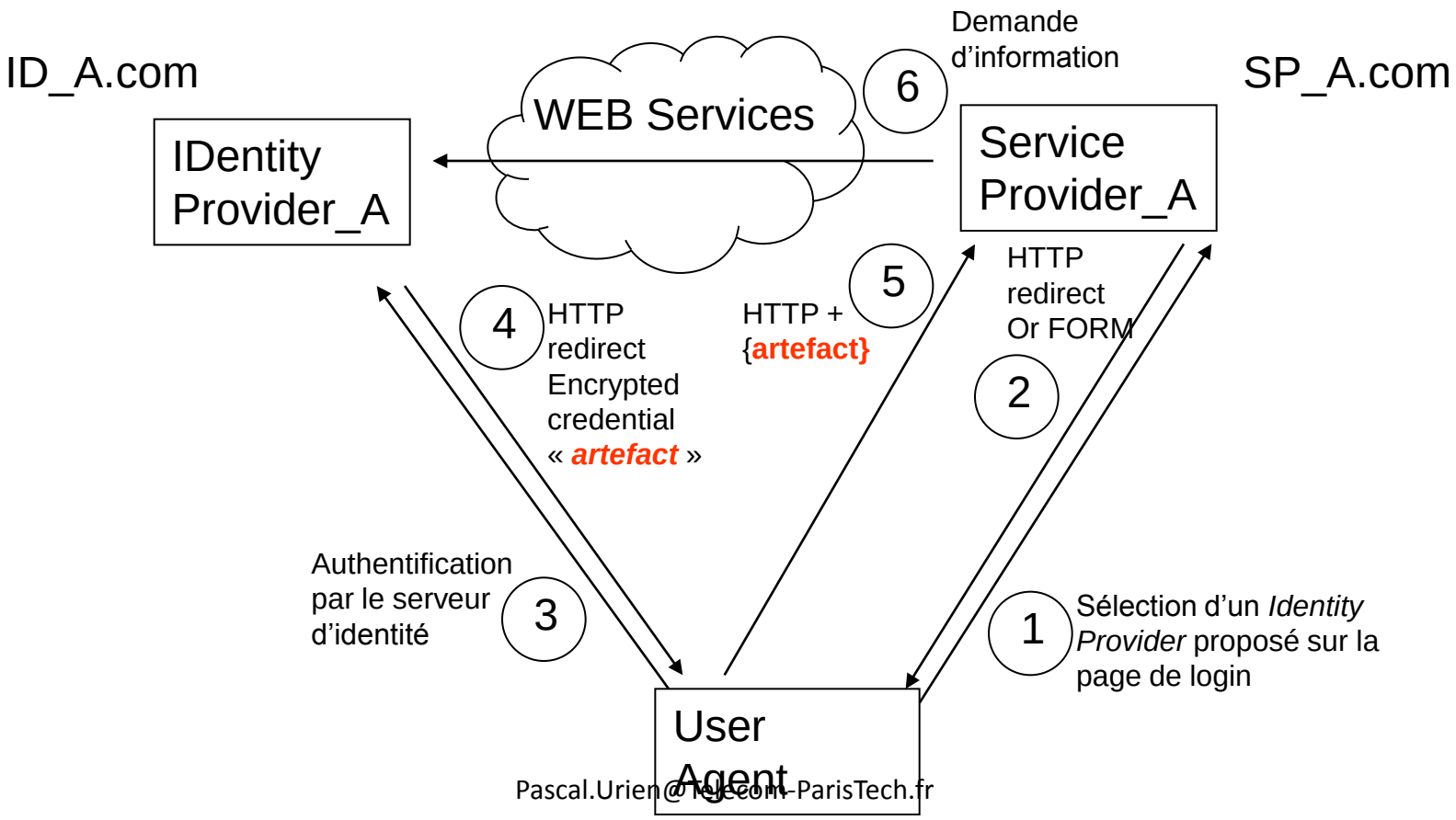
*Fédération, n. fém. (lat. foedus, foederis «alliance»).
1. Organisation politique et administrative de l'État fédéral.
2. Par ext. Groupement de sociétés, d'associations, de syndicats, etc.

2/3

- Les entités fonctionnelles
  - L'utilisateur (User); il possède un terminal informatique muni d'un navigateur WEB.
  - Les fournisseurs de services WEB (Service Provider).
    - Un Service Provider gère une identité pour déterminer/appliquer les droits/privilèges du client (Authorization).
    - "Liberty explicitly accommodates identity provider use of arbitrary authentication mechanisms and technologies"
  - Les gestionnaires d'identités (IDentity Provider).
    - La fédération de deux identités gérées par le Service Provider et Identity Provider est un acte volontaire du client.
  - Le lien entre Service Provider et Identity Provider sont réalisé par des WEB services.

et_2@ID_A.com
Alias="abcd"
- SecurityDomain= "SP_A.com"
- **Name = "1234"**

et_1@SP_A.com
Alias="1234"
- SecurityDomain= "IDP_A.com"
- **Name = "abcd"**



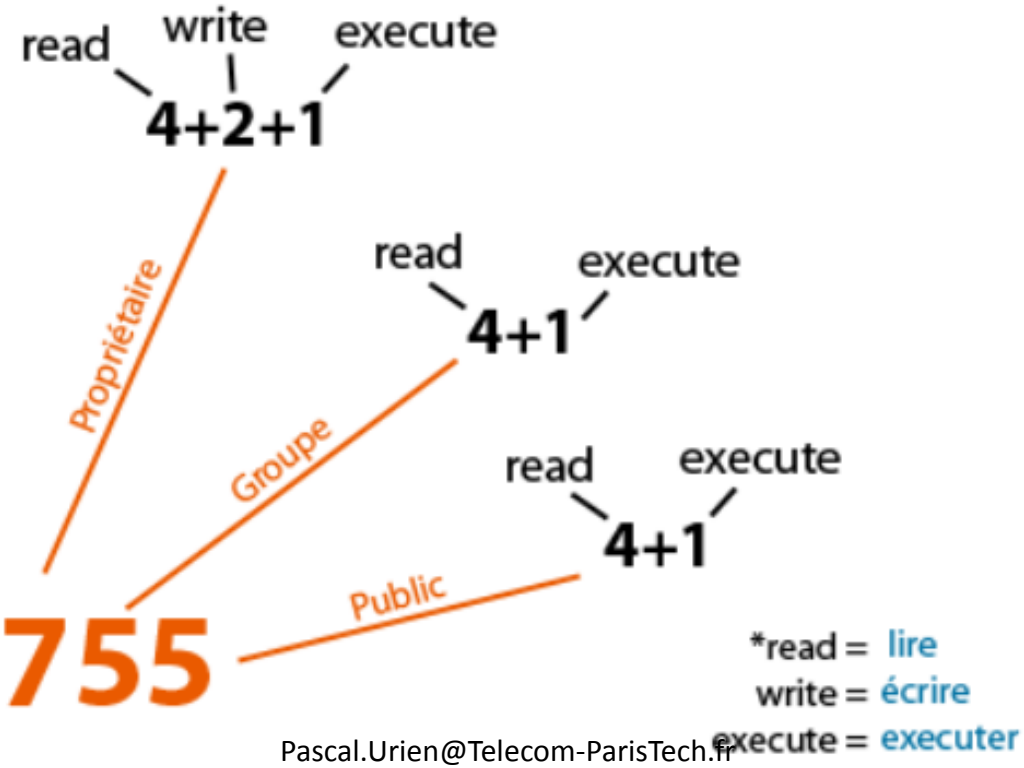
Introduction du Contrôle d'Accès

Access Control

- Mandatory Access Control MAC
  - Sous Andoid les applications sont signées par un certificat
  - Android alloue un UserID unix par certificat de signature
- Discretionary Access Control - DAC
- Role-Based Access Control - RBAC.

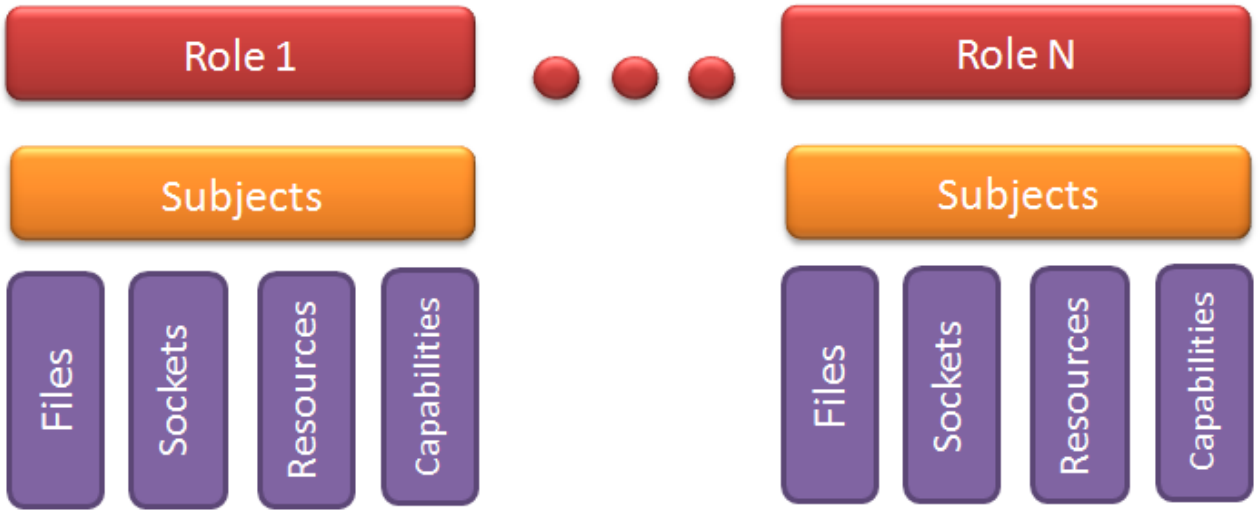
Discretionary Access Control - DAC

- Exemple
  - Gestion des droits d'accès aux fichierx sous Unix



RBAC: Role Base Access Control

- Rôle
- Sujets
- Objets



Exemple: Windows

RBAC Roles - Subjects - Objects

*PaX est un correctif (patch) de sécurité pour le noyau Linux créé en 2000

- Roles
  - Roles can be applied to a user or group
  - Everything without a specific role is given the “default” role
- Subjects
  - Subjects refer to binaries or scripts
- Objects
  - Objects are files, sockets, resources, capabilities, and PaX markings.
  - Files support access like read, write, execute, append-only, create, delete, hardlink, set suid/sgid, and hidden
    - Can also create audit logs for any of these accesses
  - Sockets can be restricted by family (inet, netlink, etc)
  - IPv4 sockets can be restricted by socket type, protocol, bind address, connect destination, and port

Kerberos

Idée Générale

- Kerberos est un protocole développé par le MIT
  - Les deux versions majeures sont v4 et v5
    - La version 4 s'appuie sur l'algorithme DES
    - La version 5 supporte 3xDES et AES
  - C'est un standard IETF, RFC 1510 (kerberos v5, 1993)
- Paradigmes
  - Utilisation de tickets établissant une preuve d'identité entre deux entités (un utilisateur et un service)
  - Génération de clés de session
- Eléments
  - Utilisateur
  - KDC, Key distribution Center
    - AS, Authentication Server
    - TGS, Ticket Granting Service
    - Base de données des clients et des clés
      - Key Version Number (kvno) est un index de clé ou de mot de passe
  - Serveur d'applications
- Structures des messages
  - Les messages sont codés selon la syntaxe ASN.1

Un peu de vocabulaire

- Realm
  - Un nom de domaine lié (example.com) à une autorité d'authentification
  - Un utilisateur appartient à un domaine si il partage un mot de passe ou une clé cryptographique avec ce dernier.
- Principal
  - Clé d'identification dans la base de donné
    - *component1/component2/.../componentN@REALM*
  - Exemples
    - user@example.com
    - Service/Hostname@REALM
      - *imap/mbox.example.com@EXAMPLE.COM*
- Ticket
  - Un ticket est délivré par le serveur d'authentification, il comporte
    - Une zone d'information chiffrée avec la clé de l'utilisateur
      - *Date Validité du ticket*
      - *Clé de session du service*
    - Une zone d'information chiffrée avec la clé du service
      - Date Validité
      - Clé de session du service

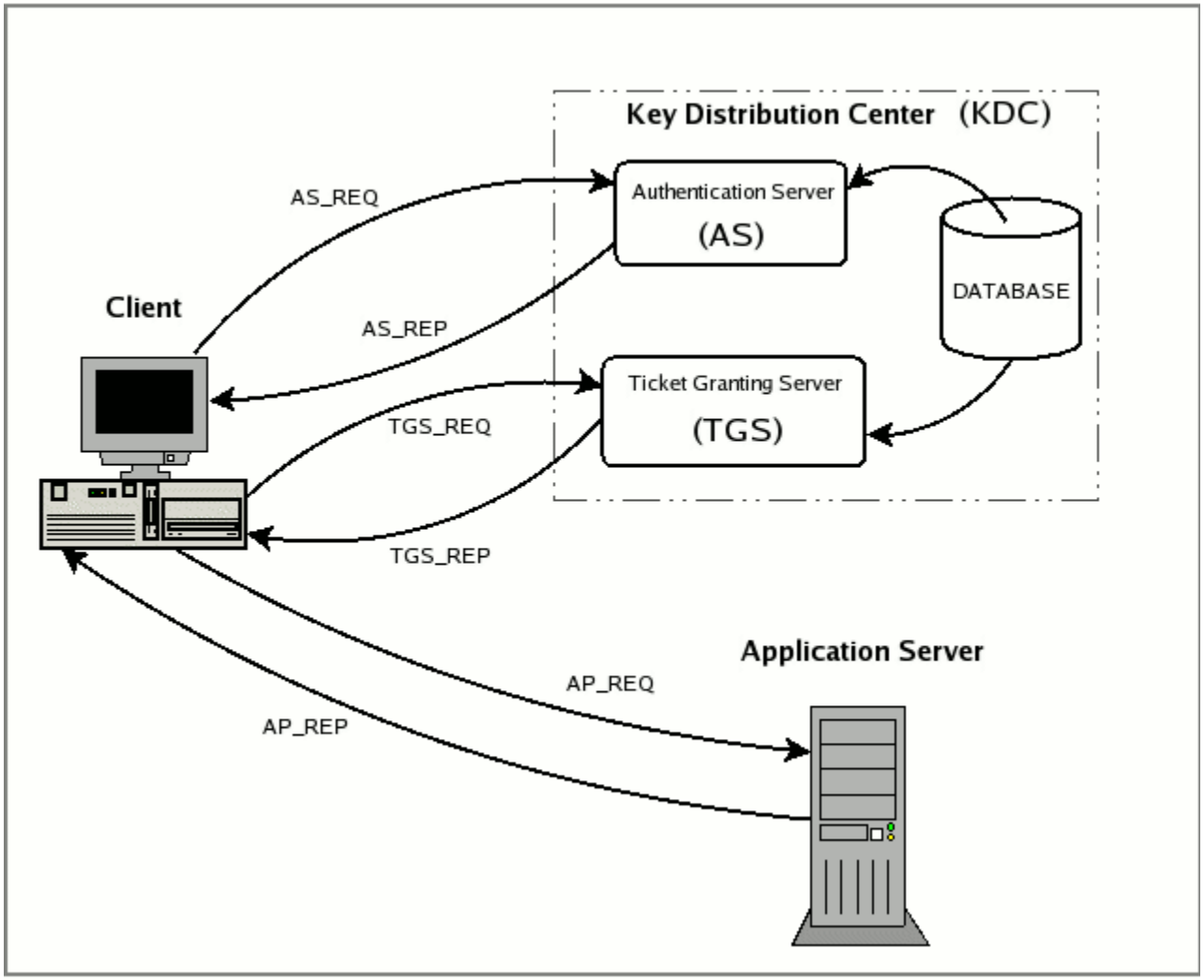
KDC et Clé de Session

- Composants du KDC
  - La base de données stocke toutes les informations associées à un principal
    - Mots de passe, clé, etc…
  - Le serveur d'authentification
    - Il réalise l'authentification d'un utilisateur basé sur son mot de passe
    - Il délivre un ticket d'authentification, Ticket Granting Ticket, ou TGT, dont le principal est krbtgt/REALM@REALM
  - Le Ticket Granting Server (TGS)
    - Il délivre des tickets de services à un utilisateur authentifié, c'est-à-dire muni d'un TGT.
- La clé de session (Session Key)
  - Pour un utilisateur c'est une clé cryptographique (DES, 3xDES, AES) déduite de son mot de passe
  - Pour un service, c'est une clé générée par le KDC

Au sujet des Authenticator

- Un authenticator est un authentifiant
  - Il est réalisé grâce au chiffrement avec la clé de session d'un ensemble de paramètres comportant l'identité de l'utilisateur et la date
- Un authenticator associé à un ticket de service évite la duplication illicite de ticket
  - Le ticket de service est chiffré avec la clé de session
  - La présence d'un authenticator « frais » prouve la connaissance de la clé de session

Architecture Kerberos



AS_REQ, AS_REP

- Authentication Server Request (AS_REQ)
  - AS_REQ = ($Principal_{Client}$, $Principal_{Service}$, IP_list , Lifetime)
    - $Principal_{Client}$ = user@REALM
    - $Principal_{Service}$= krbtgt/REALM@REALM

- Authentication Server Reply (AS_REP)
  - TGT = ($Principal_{Client}$, krbtgt/REALM@REALM , IP_list , Timestamp , Lifetime , SK_{TGS})
  - AS_REP = { $Principal_{Service}$, Timestamp , Lifetime , SK_{TGS} }K_{User} { TGT }K_{TGS}

TGS-REQ, TGS_REP

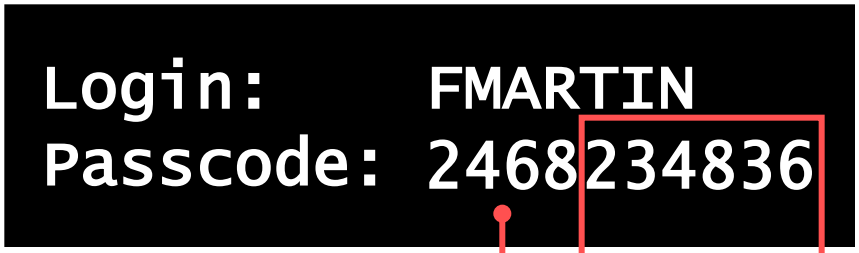
- Ticket Granting Server Request (TGS_REQ)
  - Authenticator = { $Principal_{Client}$, Timestamp }SK_{TGS}
  - TGS_REQ = ($Principal_{Service}$, Lifetime , Authenticator) {TGT}K_{TGS}
- Ticket Granting Server Replay (TGS_REP)
  - $T_{Service}$ = ($Principal_{Client}$, $Principal_{Service}$, IP_list , Timestamp , Lifetime , $SK_{Service}$)
  - TGS_REP = { $Principal_{Service}$, Timestamp , Lifetime , $SK_{Service}$} SK_{TGS} { $T_{Service}$ }$K_{service}$

AP_REQ, AP_REP

- Application Request (AP_REQ)
  - Authenticator = { $Principal_{Client}$, Timestamp }$SK_{Service}$
  - AP_REQ = Authenticator { $T_{Service}$ } $K_{service}$
- Application Response (AP_REP)
  - AP_REP = { $T_{Service}$ +1} $K_{service}$

OTP: SecureID

SecurID: Principe



Mot de passe = PIN + Token Code

Authentification forte à deux facteurs

Mot de passe dynamique, changeant toutes les minutes

10 digits ~ 33 bits

$2^{80} = 2^{10x8}$ = 24 digits =123456789012345678901234

Technologie

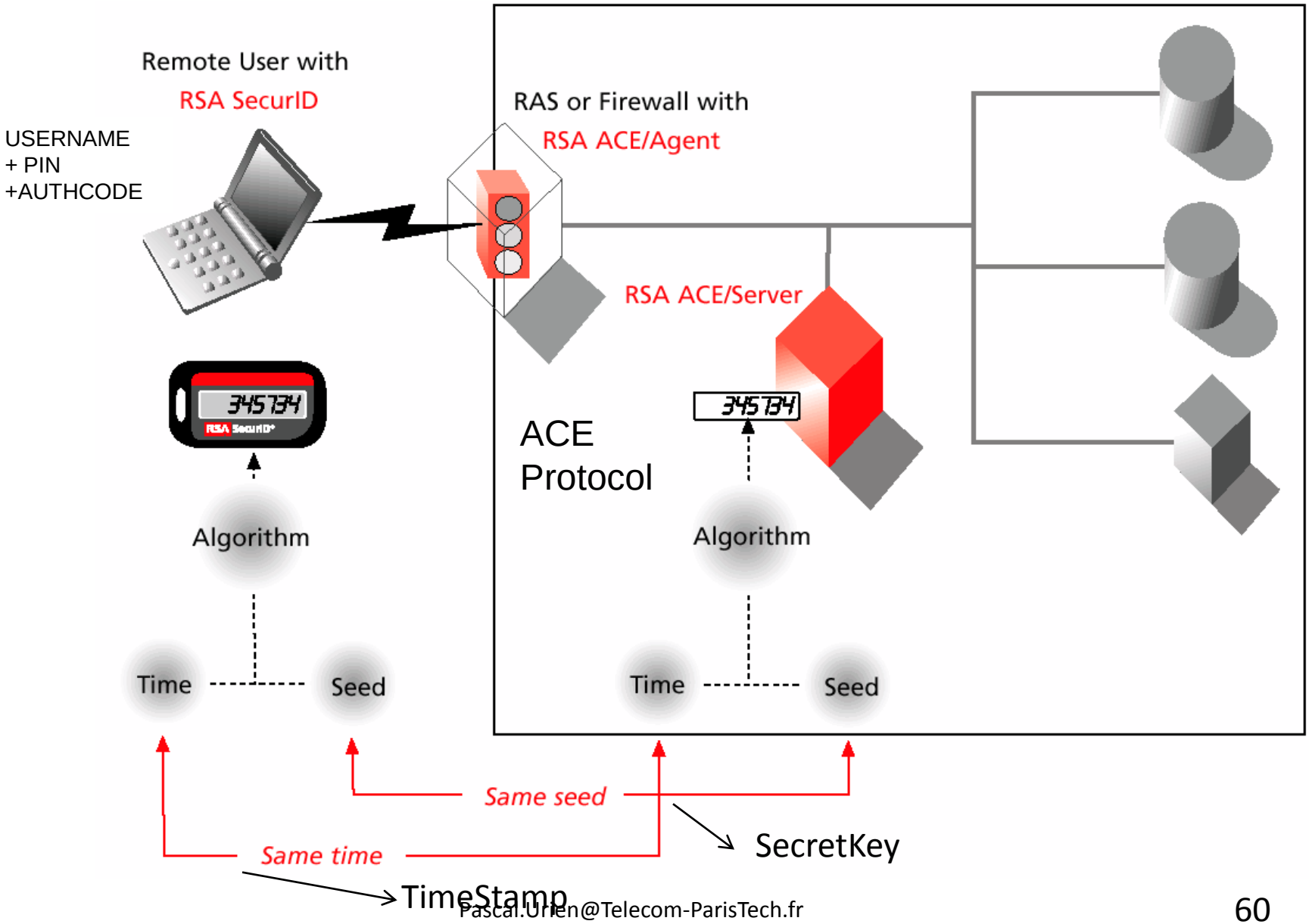
- Une fonction de hash propriétaire, h=*ASH*F
  - TokenCode = h(SecretKey || TimeStamp)
    - SecretKey, 64 bits
    - TimeStamp, 32 bits
- "Fast Software-Based Attacks on SecurID", Scott Contini and Yiqun Lisa Yin, 2003
  - *On a 2.4 GHz PC, 2^{48} hash operations take about 111 years. It would require over 1300 of these PC's to get the key in a month*
  - *Attaques publiées en 2^{40}*

$$2^{40}$$

Les 3 failles de SecureID

- Faiblesse de la clé secrète
  - Théorique de 64 bits (sécurité faible)
  - Des attaques abaissant l'entropie de la clé secrète à 40 bits ont été publiées en 2003
  - La clé peut être retrouvée en quelques heures avec 1000 PCs
- Architecture propriétaire
  - Protocole ACE.
  - Des failles publiées.
  - SOB, Secured By Obscurity.
- Protection du passcode difficile par EAP
  - L'entropie du passcode est de 33 bits seulement
  - La protection du passcode implique un secret partagé sur le terminal

Architecture ACE

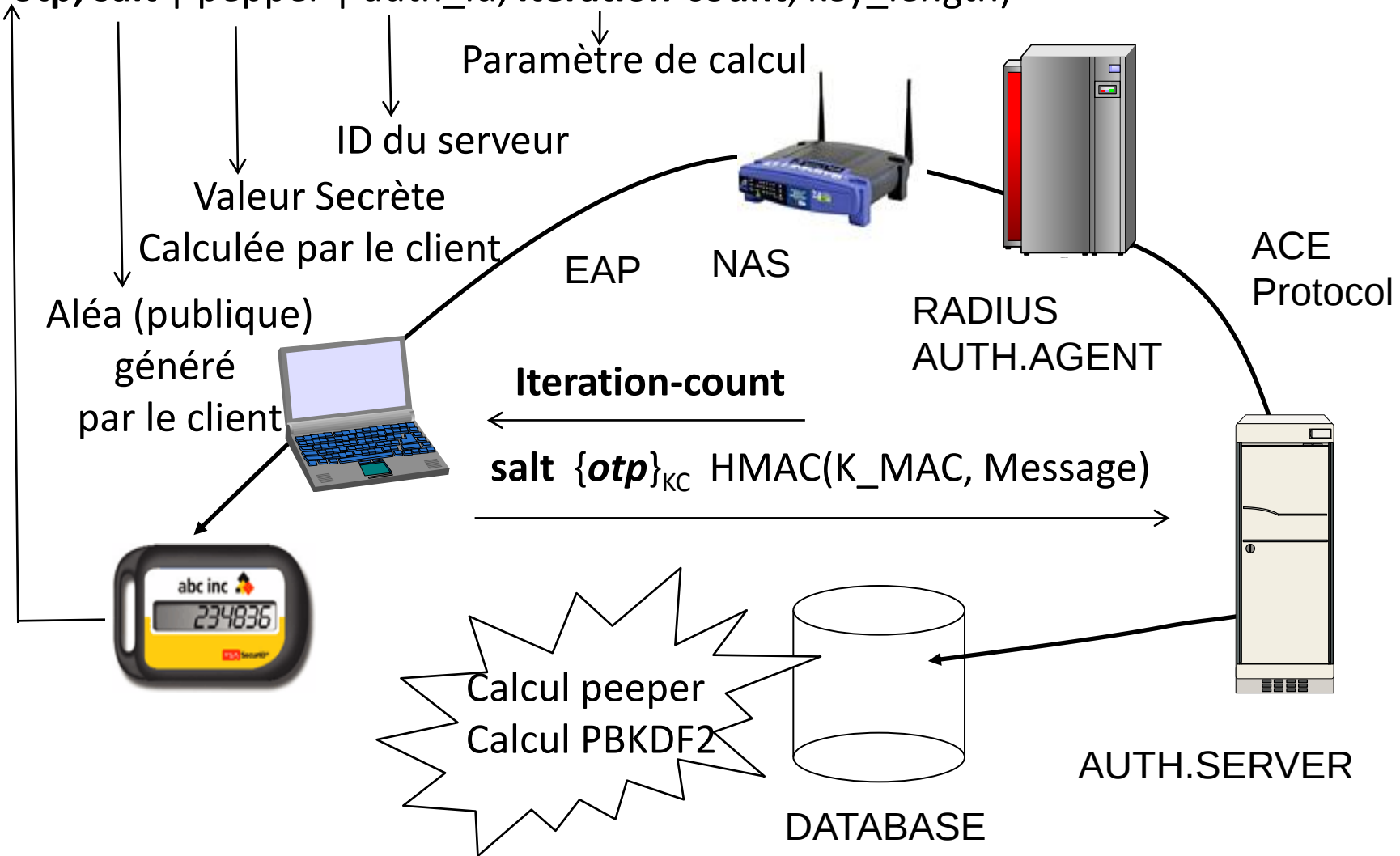


Ace 1.2.4 (1996), UDP port 124

- Le SHELL délivre un message Hello
- Le server ACE répond par un TimeStamp= T
- L'utilisateur donne son PASSCODE= Pi
- Le SHELL calcule un digest de 32 octets
  - F2($IP_{ACE\text{-}Client}$, T, Pi)
    - Soit quatre mots de 8 octets wp1, wp2, wp3, wp4
- Le SHELL génère un paquet UDP avec le login de l'utilisateur et la valeur chiffrée de wp1, {wp1}Kc, à l'aide d'une clé DES Kc,
  - La clé Kc (56 bits) a été préalablement générée et transmise au serveur ACE
  - Le serveur ACE déchiffre wp1 et calcule une ou plusieurs valeurs ($IP_{ACE\text{-}Client}$, Ti, Pi)
    - En cas de succès il retourne au client la valeur chiffrée {wp2}Kc
    - En cas d'échec il retourne au client une notification d'erreur chiffrée avec wp1, {error}wp1
  - http://www.homeport.org/~adam/dimacs.html

The EAP Protected One-Time Password Protocol , RFC 4793

K_MAC | K_ENC | MSK | EMSK | SRK =
PBKDF2(
otp, salt | pepper | auth_id, **iteration-count**, key_length)



Attaque APT

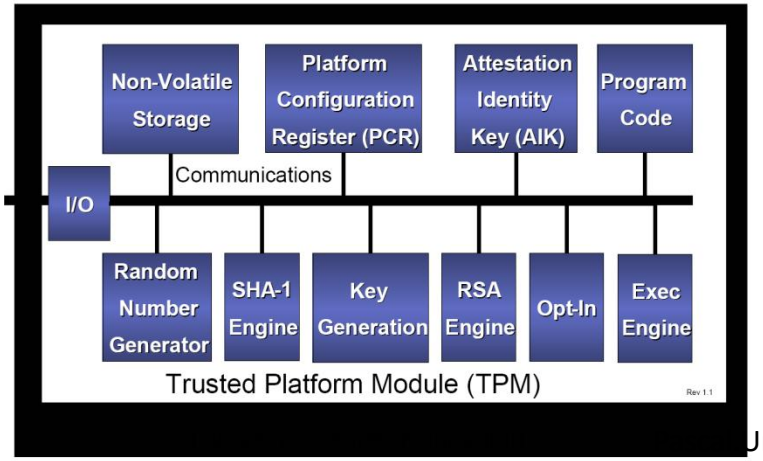
- En 2011 le système RSA SecurID a été victime d'une attaque dite APT (pour *Advanced Persistent Threat*), c'est-à-dire un malware variante du logiciel *Poison Ivy*.
- Un jeton SecurID possède un numéro de série et stocke une clé secrète de 128 bits, usuellement nommée *seed*.
- Il génère un *code* basé sur une fonction de hash, la date, et le seed, code=h(date||seed).
- L'utilisateur connait un PIN associé au jeton. Le *passcode* est la concaténation du code affiché par le jeton et du PIN utilisateur.
- Une base de données stocke les tuples *seed*, numéro de série ainsi que le login utilisateur.
- Il est possible (?) que le *seed* soit une fonction du numéro de série.
- L'attaque a (?) permis d'obtenir tout ou partie de la base de données.
- Un avertissement de la société indique qu'il serait possible de récupérer le *seed* à partir d'un seul passcode.

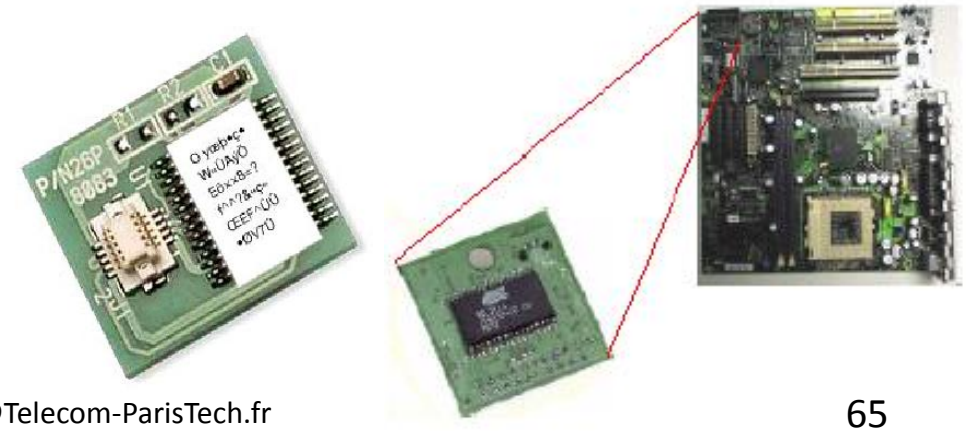
TPM

Le Trusted Platform Module (TPM)

- Un module de sécurité lié à une carte mère (TPM, *Trusted Platform Module)*
  - Un composant proche d'une carte à puce (mêmes fondeurs)
- Un modèle de sécurité figé, TCG - *Trusted Computing Group.*

Attaque	Solution courantes	Défauts	Apports TPM
Vol de données	Chiffrement, intégrité (VPN, …)	Stockage des clés dans des espaces non sures	Stockage sécurisé des données
Accès non autorisé à une plateforme	1) Login, mot de passe 2) Biométrie 3) Jetons externes (cartes à puce…)	1) Attaques par dictionnaire 2) Fiabilité des techniques biométriques 3) Crédits d'authentification indépendants de la plateforme	Protection des données d'authentification liées à la plateforme.
Accès au réseau non autorisé	Windows network logon, IEEE 802.1X	Données d'authentification stockées dans un espace non sure	Stockage sécurisé des données d'authentification.

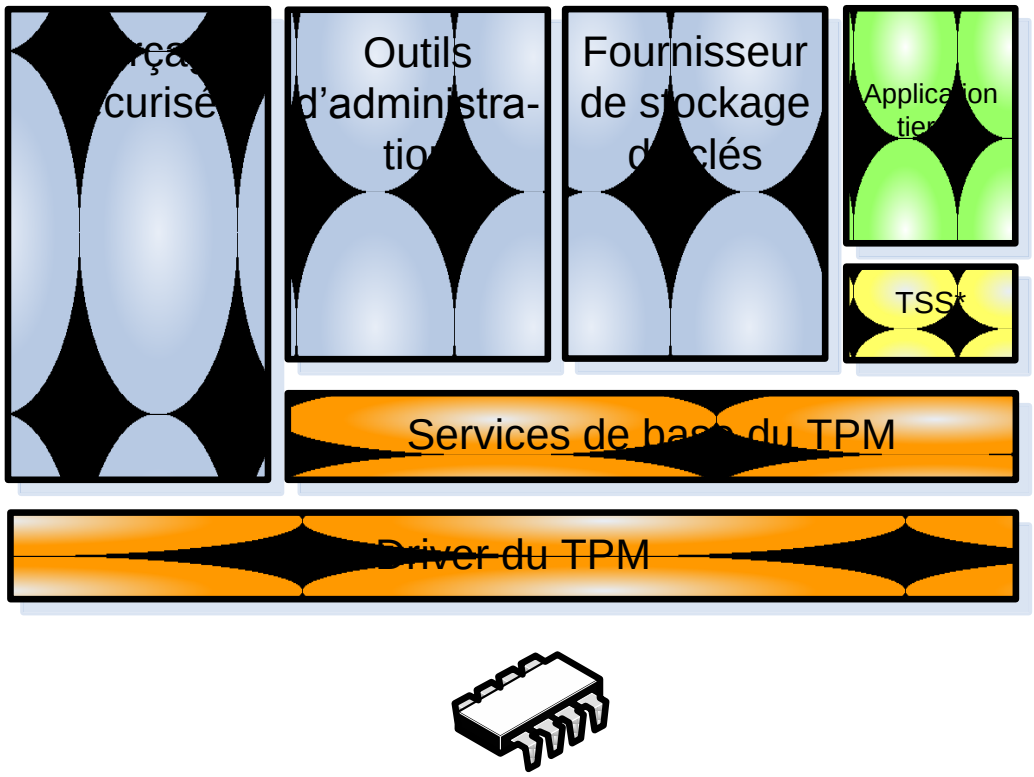


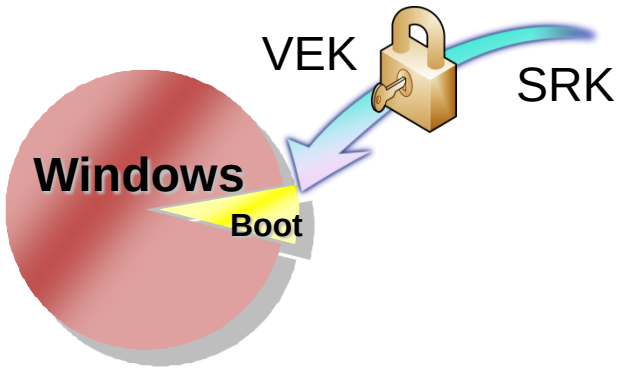


Usage des TPMs dans Windows Vista

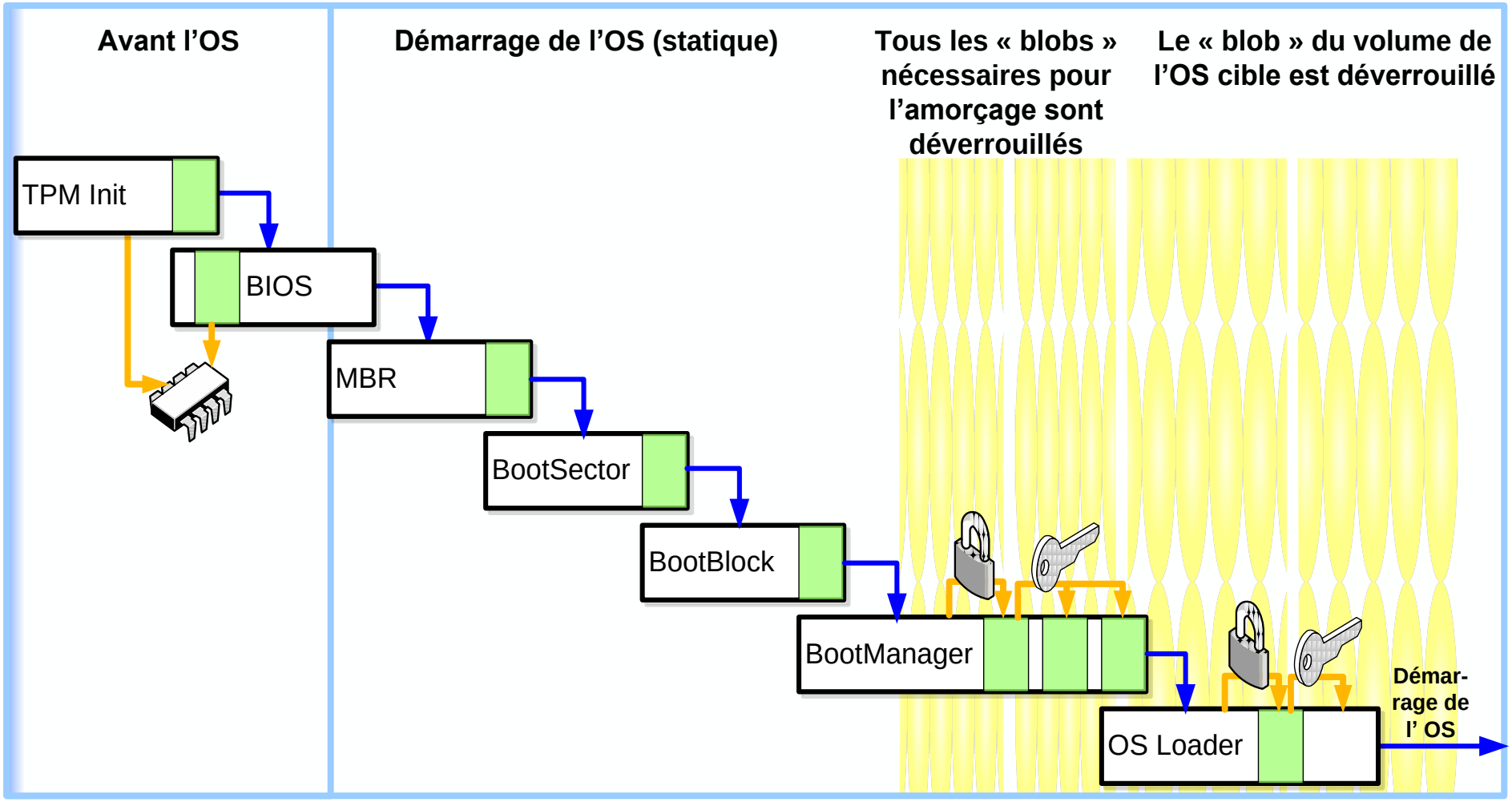
- Amorçage sécurisé (Secure Boot)
  - Contrôle de l'intégrité logicielle du système
- Chiffrement du disque (Bitlocker™)
  - La SRK chiffre une clé VEK (Volume Encryption Key)
  - La clé VEK est stockée (chiffrée par la clé SRK) sur le disque dur dans la partition de boot
  - Les PINs du TPM peuvent être gérés par une carte à puce
- Contrôle de l'intégrité des PCs au moment de la connexion réseau (NAP, Network Access Protection)
  - Le protocole NAP permet de mesurer l'intégrité d'un nouveau nœud du réseau. Il est associé à des mécanismes de politique d'accès et de quarantaine.
  - Le protocole NAP (propriétaire) est transporté par EAP ou IPSEC

TPM


Outils
Fournisseur
Application
TSS*
Services de base du TPM
Driver du TPM


VEK
SRK
Windows
Boot

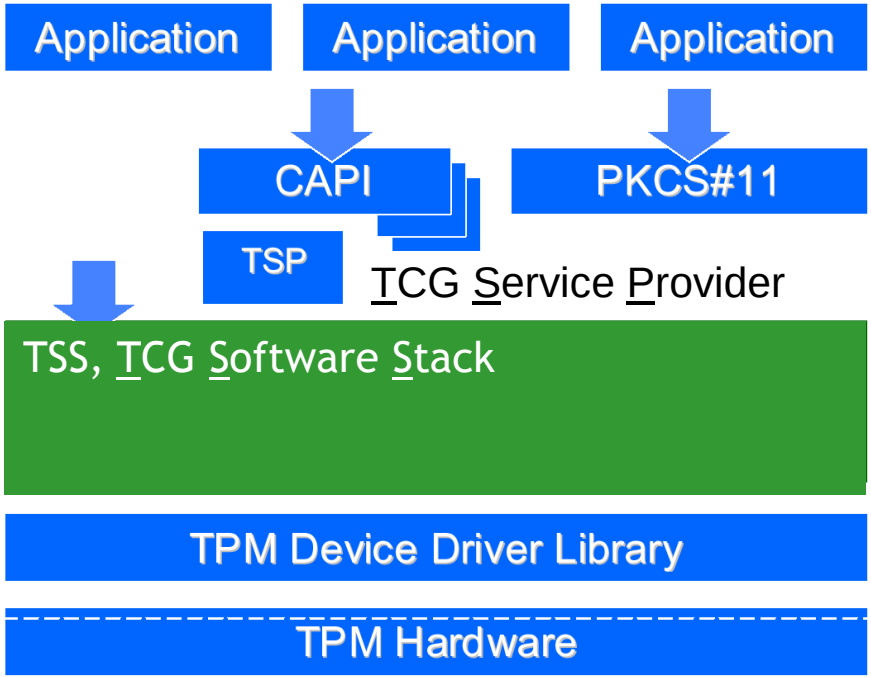
TPM

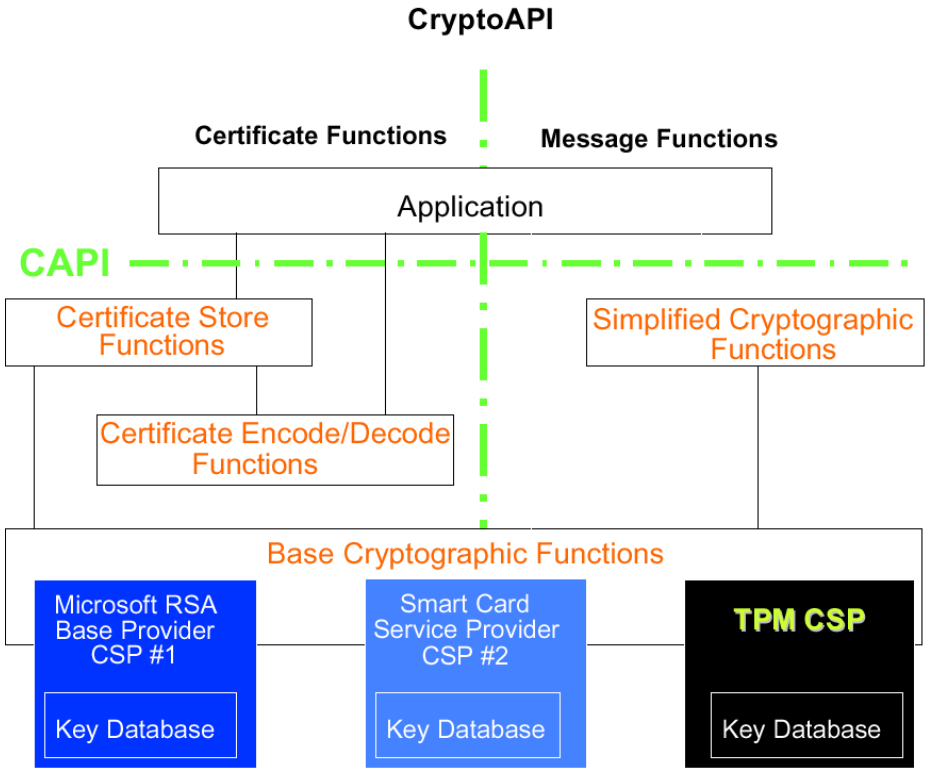

Avant l'OS
Démarrage de l'OS (statique)
Tous les « blobs » nécessaires pour l'amorçage sont déverrouillés
Le « blob » du volume de l'OS cible est déverrouillé
TPM Init
BIOS
MBR
BootSector
BootBlock
BootManager
OS Loader
Démar-rage de l' OS

TCG, Modèle

- Owner Key, clé maître d'autorisation symétrique, 160 bits
- Endorsement Key (EK),clé RSA 2048 bits
  - Deux modes de génération sont disponibles, création par le TPM, ou stockage sous contrôle de la clé Owner Key.
  - Permet d'établir un canal de communication sure {M}EK-publique avec le TPM (qui stocke la clé EK-privée).
- Storage Root Key (SRK), clé RSA 2048 bits
  - Le TPM gère un anneau de clés privés RSA de stockage; la clé d'indice 0 est SRK; la clé d'indice k est chiffrée par la clé d'indice k-1. Une goutte de clé (key blob) est la valeur chiffrée d'une clé K stockée à l'extérieur du TPM.
  - Une goutte d'information (data blob) est un fichier protégé à l'aide d'une clé de stockage K.
  - La hiérarchie des clés de stockage est par exemple TPM, administrateur, utilisateur.
  - Les clés de stockage sont activées à l'aide de Personal Identification Number (PIN, mot de passe, biométrie, carte à puce…)
- Platform Configuration Register (PCR)
  - 16 registres de 20 octets sont disponibles pour stocker des empreintes (produite par un fonction SHA1)
  - Une goutte d'information peut être liée au contenu d'un registre PCR, dans ce cas les données sont liées à la plateforme
- Attestation Identity Keys (AIKs)
  - Le TPM stocke des clés publiques délivrées par une autorité de certification CA. L'action conjointe de CA et de l'administrateur du TPM ({{M}AIK-Private)}EK-publique) permet de produire des données d'authentification pour un service (par exemple une paire de clé RSA).
- Des clés RSA 2048 bits
  - Les clés Parents (Parent Keys) sont crées sous contrôle de la clé Owner Key.
  - Les clés Enfants (Child Keys) sont générées sous le contrôle d'une Clé parent.

Pile Logicielle TCG

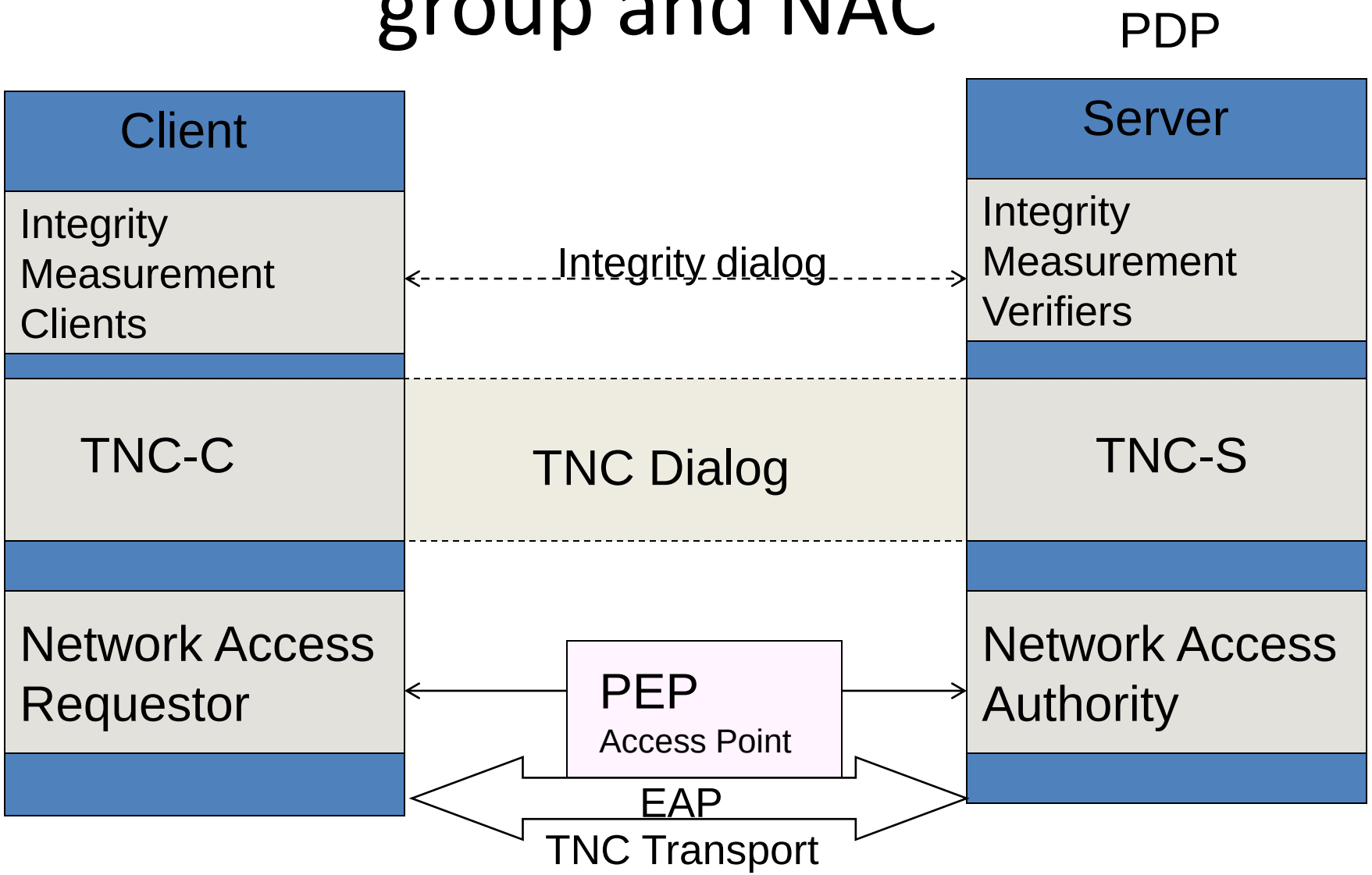




What is Network Access Control (NAC) ?

- Controlling access to the network based on platform configuration state
  - Anti-virus
  - Firewall configuration
  - Approved applications
  - Etc…
- Authenticating platform or user or both

Trusted Network Connect (TNC) sub group and NAC



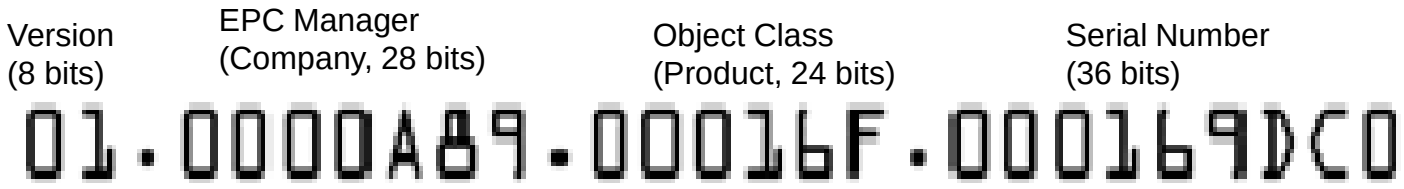
RFID

Etiquettes Electroniques 1/3

- Les produits sont actuellement identifiés par des codes barres à lecture optique.
  - GTIN, Global Trade Item Number (code à 8,12,13,14 chiffres)

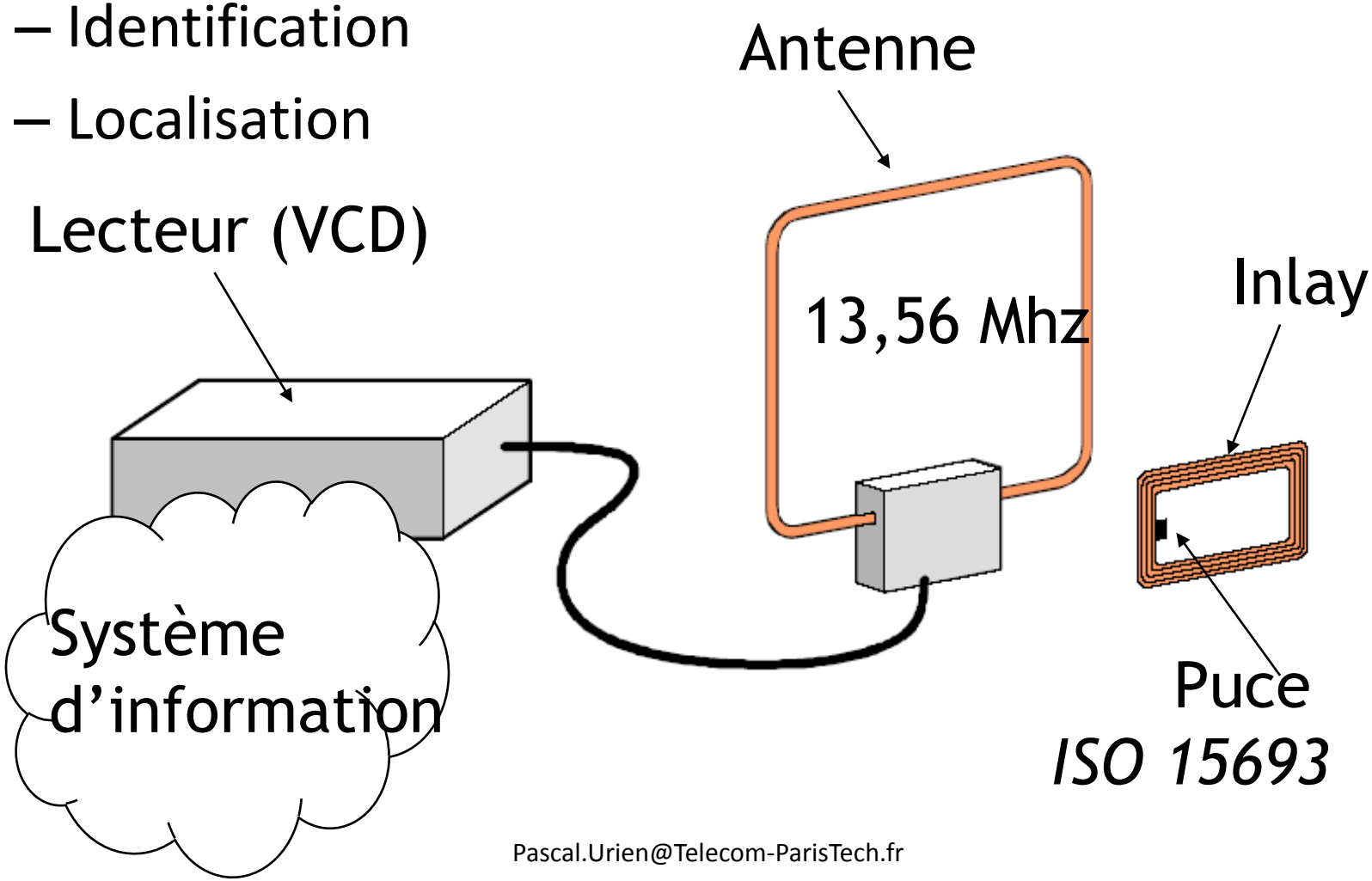
Exemple: Code GTIN, 13 chiffres

- Etiquette électronique (TAG)
  - EPC, Electronic Product Code (code à 64, 96 bits)



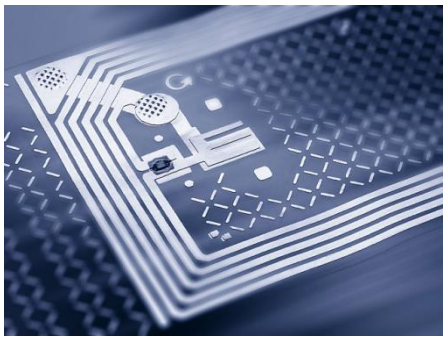
RFID 2/3

- Services
  - Identification
  - Localisation



RFID ISO-15693 3/3

- Vicinity coupling device (VCD), Vicinity Integrated Circuit Card (VICC).
  - Fréquence de travail (fc) 13,56Mhz.
  - Débits (en mode single carrier) 6,62 kbits/s (fc/2048) 26,48 kbits/s (fc/512)
  - Distance de fonctionnement compris entre quelques centimètres et un mètre.
  - Champs magnétique d'opération compris entre 0,15 A/m et 5 A/m
- Organisation de la mémoire
  - 256 blocs au plus de taille maximale 256 bits
- Paramètres VICC
  - Unique identifier (UID) 64 bits.
  - Application family identifier (AFI) 8 bits.
  - Data storage format identifier (DSFID) 8bits.



- 15 Commandes VICC sont définies par la norme ISO 15693-3
  - Inventory, Stay Quiet, Reset to Ready, Select
  - Read Single Block, Read Multiple Blocks, Write Single Block, Write Multiple Blocks, Lock Block, Get Multiple Block Security Status
  - Write AFI, Lock AFI
  - Write DSFID, Lock DSFID
  - Get System Information
  - Custom, Proprietary.

Identity Based Encryption
IBE
Le Futur des Réseaux ?

Pairage bilineaire

- Soit G1, G2, et Gr des groupes finis d'ordre r, avec r premier
- Un pairage bilinéaire e (*bilinear pairing, bilinear map*) est une fonction telle que
  - e: G1xG2 -> Gr
  - e est non dégénérée, e(P,Q) # 1
  - e est bilinéaire $e(aP, bQ) = e(P,Q)^{ab}$

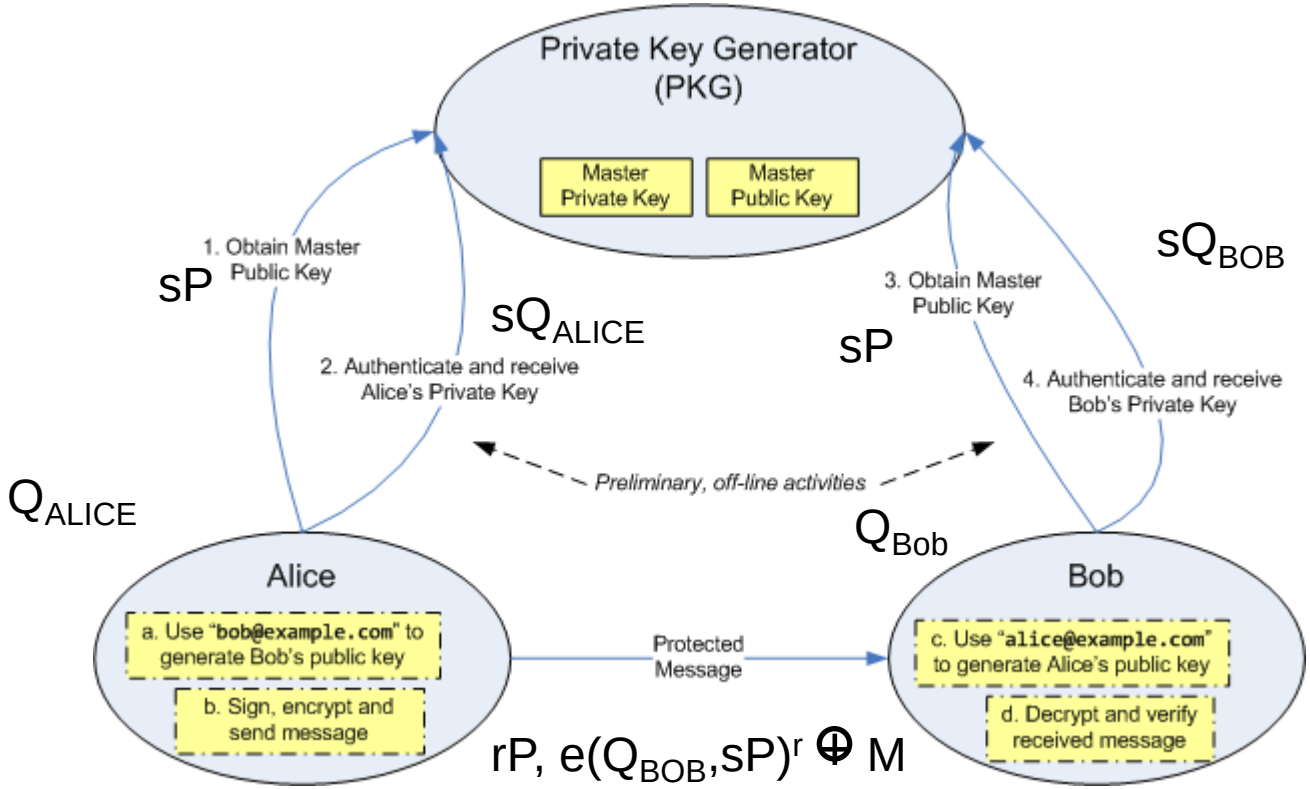
Groupe de r-torsion

- E(K) une courbe elliptique définie sur un corps K (Fp), de caractéristique q ($p=q^m$)
  - Les points de E peuvent être définis (il existe un isomorphisme) sur des corps plus grands Fq^k , ou encore la clôture algébrique de $\overline{K}$.
- Pour r entier, le sous groupe de points de r-torsion, défini pour une courbe elliptique E sur un corps K, est noté
  - $E(K)[r] = \{ P \in E \mid rP = O\}$
  - Pour r premier avec q (soit pgcd(r,q)=1)
    - E(K)[r] est isomorphe à Z/rZ x Z/rZ
    - E(K)[r] contient r^2 points
- Si r divise p-1 (q^m - 1), le corps K (Fp) contient les racines $r^{ième}$ de l'unité

Couplage de Weil

- Couplage de Weil, E(K) courbe elliptique sur le corps K
  - Corps K =Fq
  - e: E(K) [r] x E(K) [r] -> Um
    - $e(aP,bQ)= e(P,Q)^{ab}$
  - Um est le groupe des racines $r^{ième}$ de l'unité dans Fq^k
    - $Um = \{ x \in Fq^k \mid x^r = 1 \}$
    - r divise le cardinal de E(K) (# E(K))
    - r divise $q^k - 1$
  - Le premier algorithme de couplage de Weil a été proposé en 1986 par *V. Miller*, "Short Programs for Functions on Curves"

Architecture IBE



e: GxG -> Gr

IBE

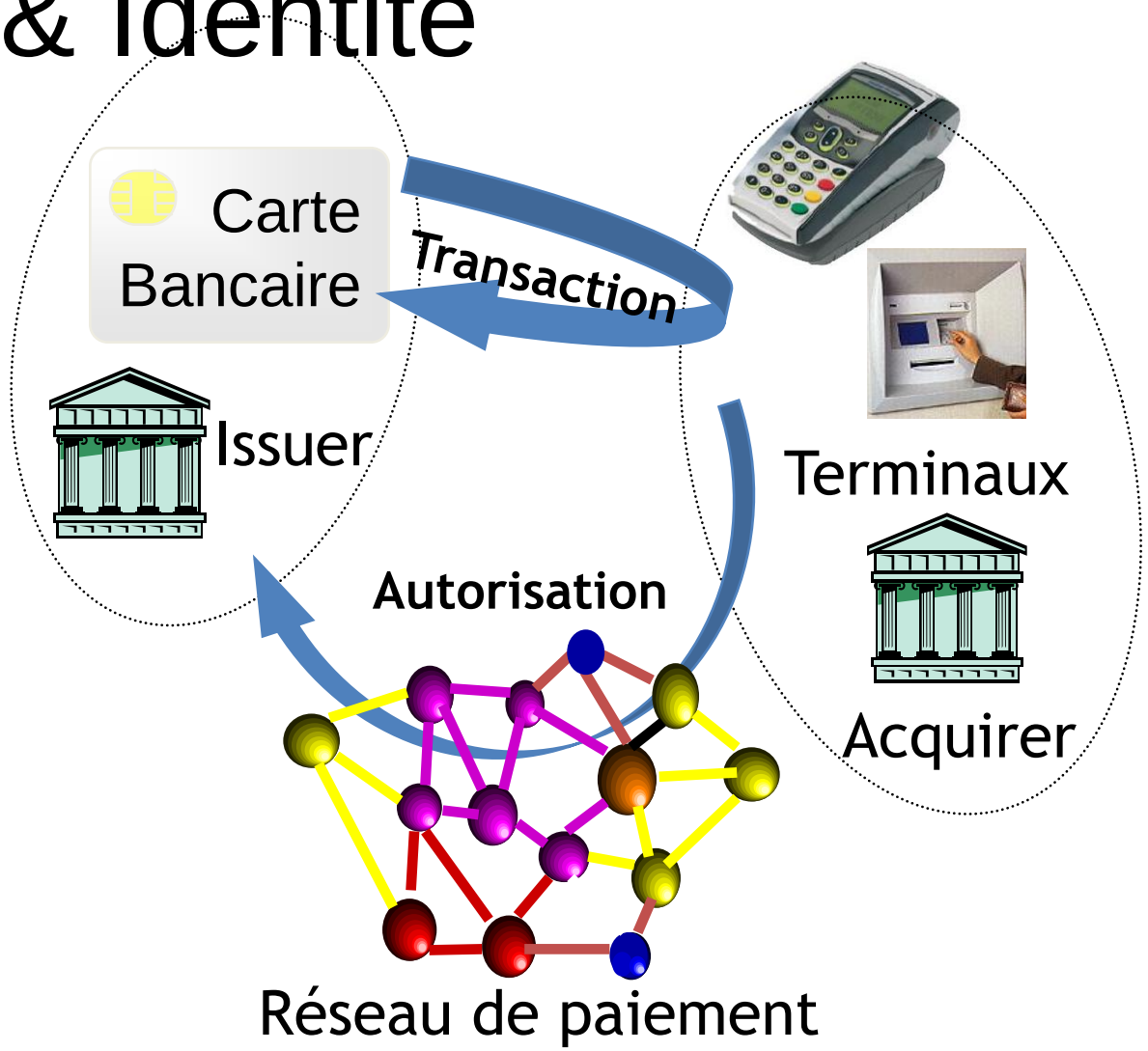
- DH Tripartie
  - Trois clés publiques aP, bP, cP
  - Trois clés privées a,b,c
  - $e(aP,bP)^c = e(bP,cP)^a = e(bP,cP)^a$

- IBE (Identity Based Encryption) version de base
  - P un point d'ordre n
  - Clé maitre (privée) s Є [0, n-1]
  - Clé publique sP
  - H: {identités} -> G
  - Q_{ID} = H(ID), par exemple Q_{ID} = h(ID).P', clé publique de ID
  - s Q_{ID} Clé privé de ID
  - r un nombre aléatoire dans [0, n-1]
  - Chiffrement à l'aide de la clé publique sP, et d'un nombre aléatoire r
    - Secret = $e(Q_{ID}, sP)^r$,= $e(Q_{ID},P)^{s\ r}$ C = M exor Secret
  - Transmission (Q_{ID},rP, C)
  - Déchiffrement S= $e(sQ_{ID},rP) = e(Q_{ID},P)^{s\ r}$, M = C exor S

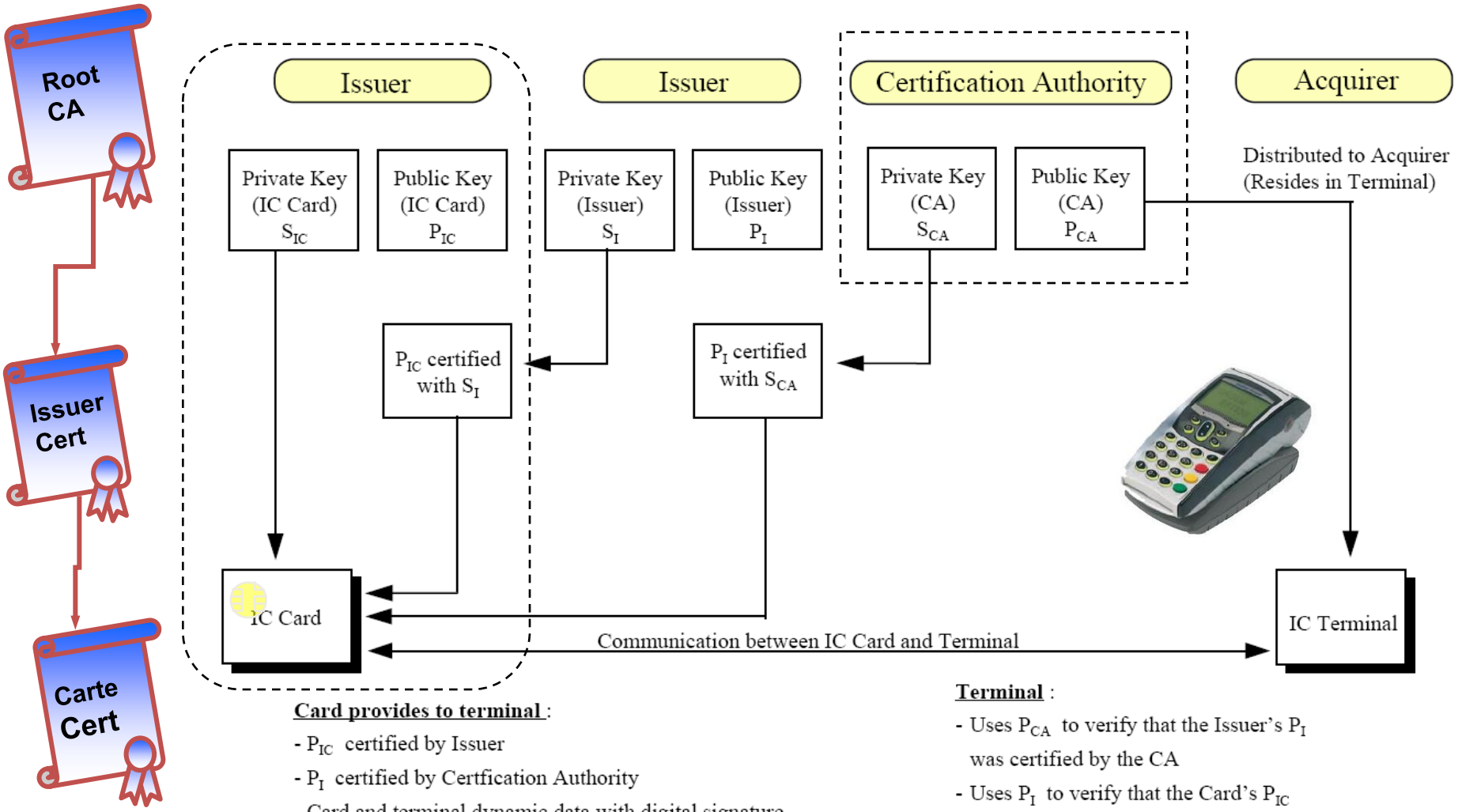
Identité Bancaire

Infrastructure de paiement EMV & Identité

- Trois entités fonctionnelles
  - Emetteur de la carte.
  - Acquéreur.
  - Réseau.
- Identité
  - Numéro de carte (PAN)
- Deux types d'authentification
  - Statique, valeur signée stockée dans la carte
    - Certificat EMV
  - Dynamique, algorithme cryptographique exécuté par la carte
    - Clé privée RSA
- Authentification à deux facteurs
  - PIN code du porteur
  - Clés cryptographiques gérées par l'émetteur
    - RSA et certificats
    - 3xDES pour la génération de cryptogrammes de paiement



EMV: Dynamic Data Authentication



Card provides to terminal :

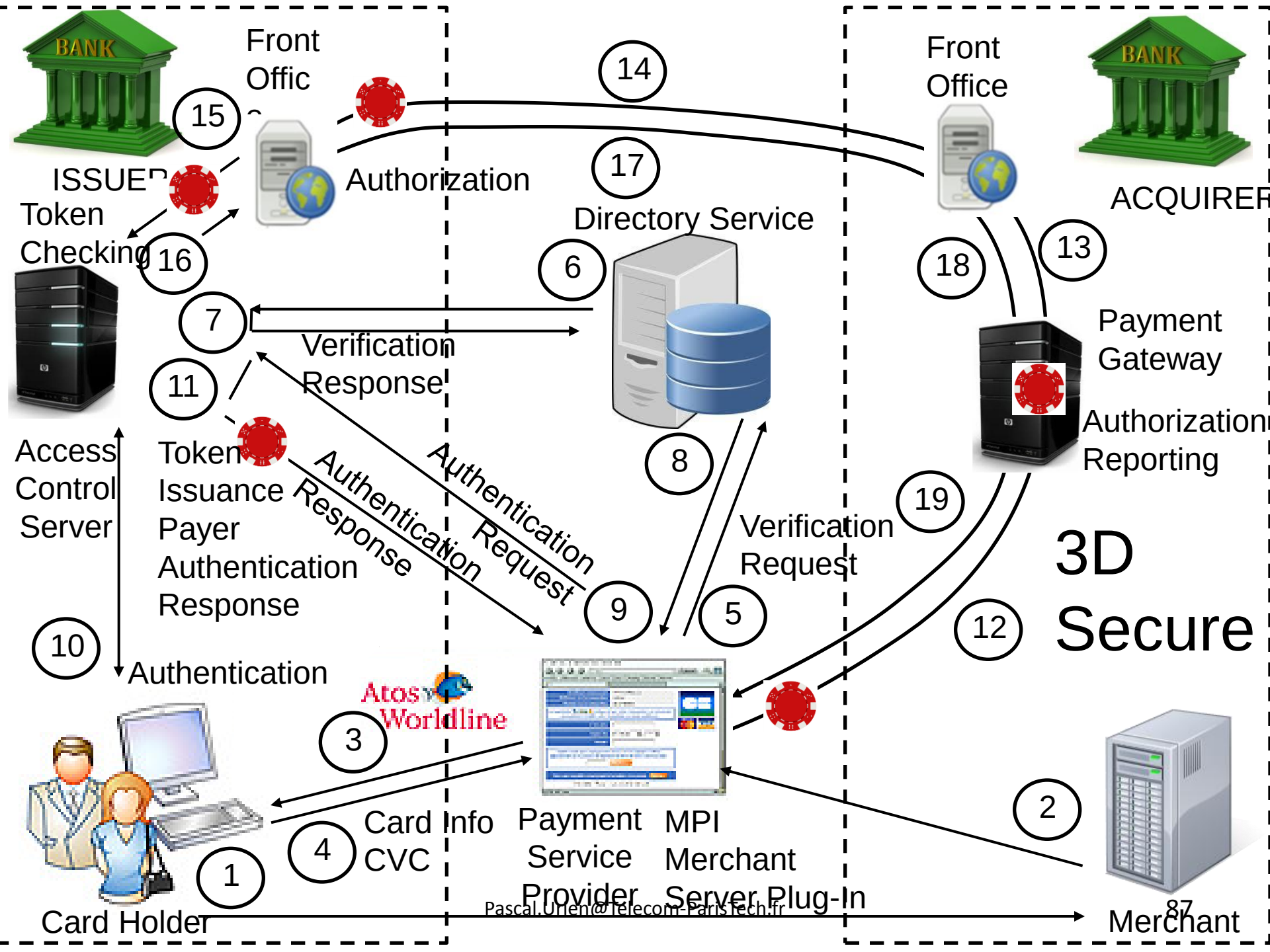
- P_{IC} certified by Issuer
- P_I certified by Certfication Authority
- Card and terminal dynamic data with digital signature

Terminal :

- Uses P_{CA} to verify that the Issuer's P_I was certified by the CA
- Uses P_I to verify that the Card's P_{IC} was certified by the Issuer
- Uses P_{IC} to verify the digital signature of the card data

Exemple: ARQC

- >> 80AE8000 1D
  - 00 00 00 00 00 00, the transaction amount
  - 00 00 00 00 00 00, the cash back
  - 00 00, the national code of the payment terminal
  - 80 00 00 00 00, the terminal verification result
  - 00 00, the transaction currency code
  - 01 01 01, the transaction date
  - 00, the type of transaction
  - 12 34 56 78, a four bytes random value
- << 77 1E
  - 9F 27 01 80, Cryptogram Information Data
  - 9F 36 02 00 18, Application Transaction Counter (ATC)
  - 9F 26 08 80 29 D3 A0 BB 2A 5E 60, Application Cryptogram
  - 9F 10 07 06 7B 0A 03 A4 A0 00, Issuer Application data


BANK
Front Offic
ISSUER
Token Checking
Authorization
14
15
16
17
Directory Service
6
7
Verification Response
11
Token Issuance Payer Authentication Response
Authentication Response
Authentication Request
Access Control Server
10
Authentication
8
9
5
Verification Request
Front Office
BANK
ACQUIRER
18
13
Payment Gateway
Authorization Reporting
19
12
3D Secure
Atos Worldline
3
4
Card Info CVC
1
Card Holder
Payment Service Provider
MPI Merchant Server Plug-In
2
87
Merchant

Techniques Biométriques

Techniques Biométriques

- De multiples modalités
  - Empreinte digitale, empreinte d'iris, empreinte palmaire, reconnaissance faciale …
- La biométrie n'est pas exacte, il y a un risque statistique. :
  - Taux de rejet (**FTE** ou *Failure To Enroll*): pourcentage d'échantillons non analysés en raison de défauts.
  - Taux de faux rejets (**FRR** ou *False Rejection Rate*) : . pourcentage de non reconnaissance erronées.
    - Pose un problème d'indemnisation d'un éventuel préjudice.
  - Taux de fausses acceptations (**FAR** ou *False Acceptation Rate*) : pourcentage de reconnaissances erronées.

- Alternative au mot de passe
  - Remplacement du PIN code d'une carte à puce (par exemple *DeXa Badge* d'Axalto).
  - Remplacement des mots de passe (PC + TPM)

Passeport électronique, carte d'identité
  - Authentification à deux facteurs
    - Techniques d'imprimerie sécurisées
    - Carte à puce (avec ou sans contact)
      - empreinte digitale
      - photographie

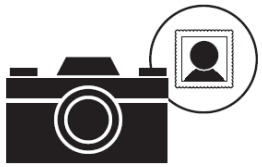
Base de données centralisée ou distribuée
  - BD distribuée: données personnelles
    - Technologie *Match On Card (MOC)*
    - Java Card Biometry API
  - BD centralisée: programme US-Visit
    - Lecture du passeport
    - Enregistrement des empreintes des index.
    - Photographie du visage

© actronix

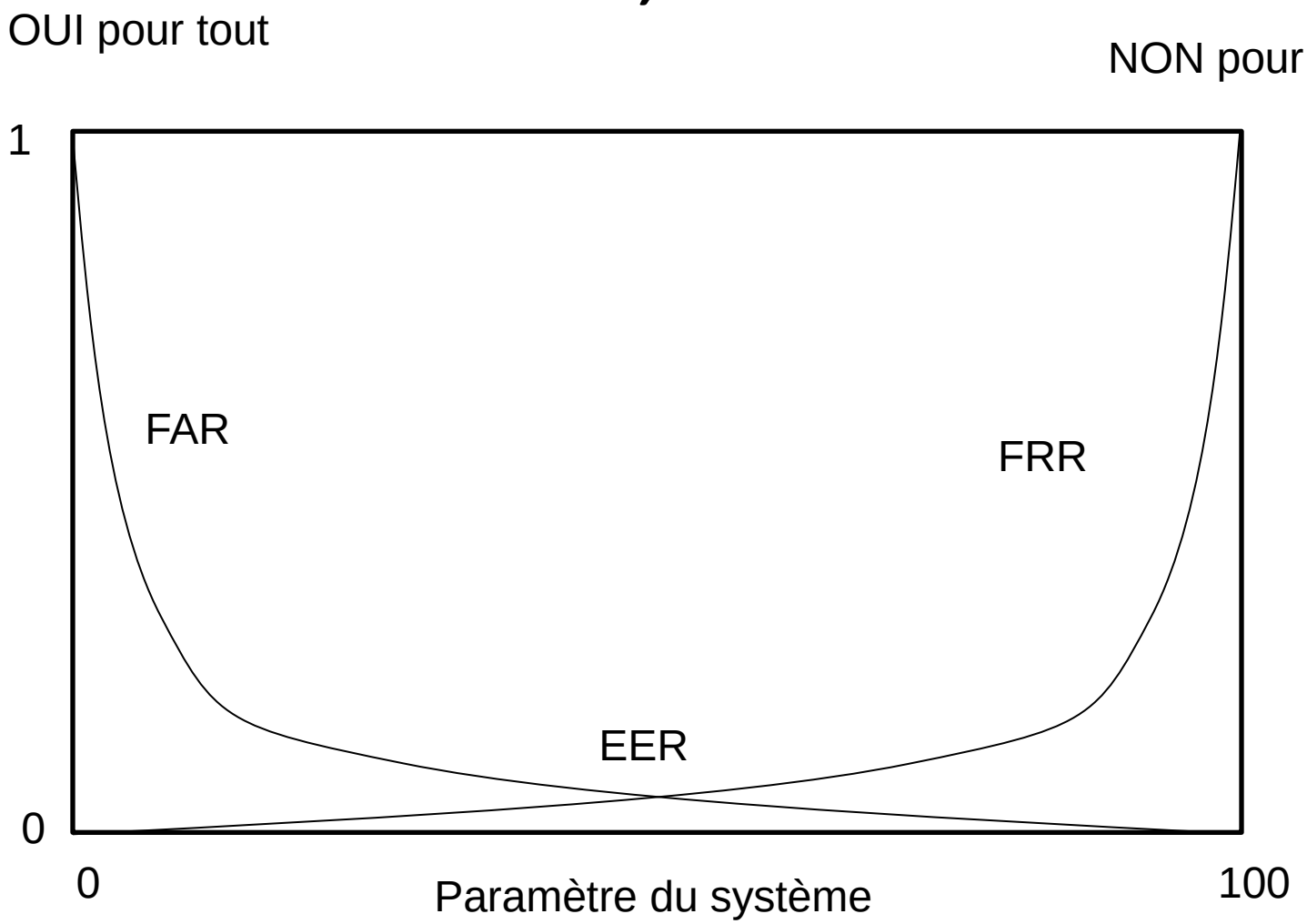
Caractéristiques :
• Jusqu'à 4000 empreintes mémorisées au sein d'un même boîtier permettant les opération d'enrôlement et de vérification
• Interface utilisateur intuitive avec 3 témoins lumineux et signaux sonores.
• Détection automatique du doigt permettant une identification simple et instinctive.
• Boîtier compact et ergonomique.
• Options multiples d'administration :
- en autonome (en connectant un PC portable)
- en réseau (option Ethernet)
Temps de vérification : < 1 seconde pour 100 utilisateurs enrôlés
Temps d'enrôlement : < 3 secondes

Taux :		
Fausse Acceptation	Rejet	d'Egal Erreur (EER)
0,005 %	0,01 %	0.1%

Prix : environ 1 400 €

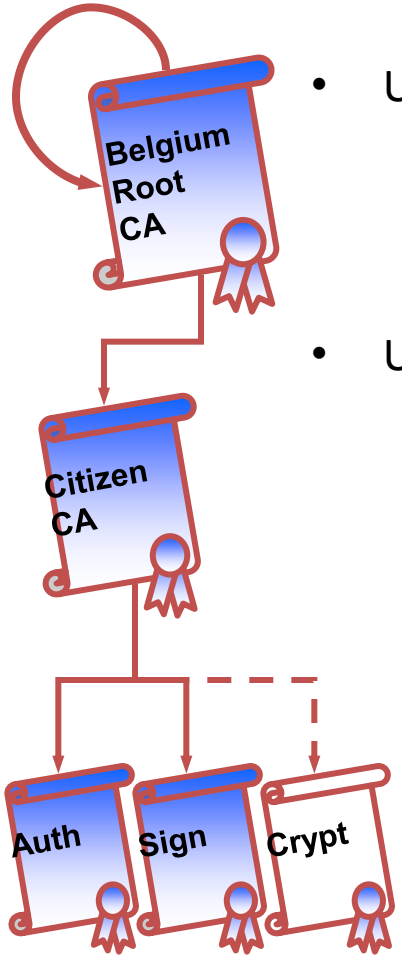


FAR, FRR


OUI pour tout
NON pour tout
1
0
FAR
FRR
EER
0
100
Paramètre du système

La carte d’identité Belge

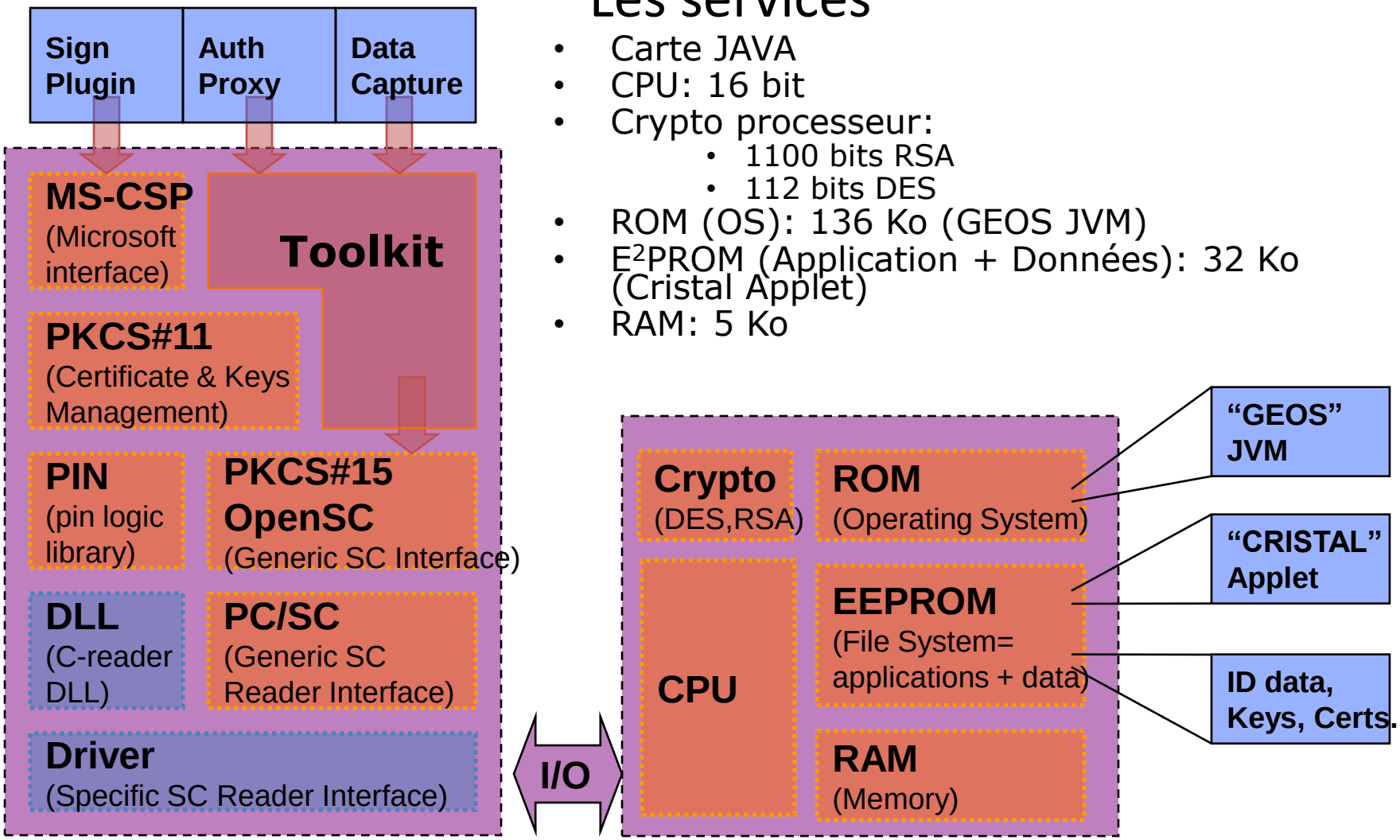
La carte d'identité Belge: bi facteurs



- Une carte imprimée
  - Imprimerie sécurisée (rainbow and guilloche printing, Changeable Laser Image –CLI-, Optical Variable Ink –OVI-, Alphagram, Relief and UV print, Laser engraving)
  - Nom, Prénom, Genre, Date de naissance, Nationalité, numéro de carte, période de validité, signature, photographie
- Une puce tamper resistant
  - Une carte java de 32 Ko E2PROM qui loge application spécifique (eID)
  - Un répertoire (Dir-ID) qui stocke des données (nom, prénom, adresse, photographie) identiques à celles qui sont imprimées et signées par le RRN (Rijksregister – Registre National).
  - Un répertoire (Dir-BelPIC, protégé par PIN code)), conforme au standard PKCS15, qui stocke des certificats X509 émis par le Citizen CA, et les clés RSA privées.
    - Un certificat d'authentification (Auth), authentification WEB, SSO…
    - Un certificat de signature (Sign), document, formulaire WEB, …
    - Un certificat (Crypt) pour les opérations de chiffrement est également prévu

Les services

- Carte JAVA
- CPU: 16 bit
- Crypto processeur:
  - 1100 bits RSA
  - 112 bits DES
- ROM (OS): 136 Ko (GEOS JVM)
- E²PROM (Application + Données): 32 Ko (Cristal Applet)
- RAM: 5 Ko



Le Passport Electronique

Le Passeport Electronique

- Le passeport électronique est décrit par les normes ICAO 9303 (part 1,2,3)
- L'application passeport gère un ensemble de fichiers
  - EF.COM, la liste des fichiers stockés dans le passeport
  - EF.DG1, la copie des informations imprimées dans le MRZ (Machine Readable Zone)
  - EF.DG2, contient une photo biométrique du propriétaire du passeport
  - EF.DG3 contient les empreintes digitales. Le contenu est chiffré ou protégé par une procédure d'authentification nommée EAC (Extended Access Control)
  - EF.DG11 diverses informations additives sur le propriétaire du passeport
  - EF.DG12 diverses informations additives sur le passeport
  - EF.DG15 stocke la clé publique (RSA) optionnelle utilisée par le mode AA (*Active Authentication)*
  - EF.SOD contient la signature du contenu du passeport.

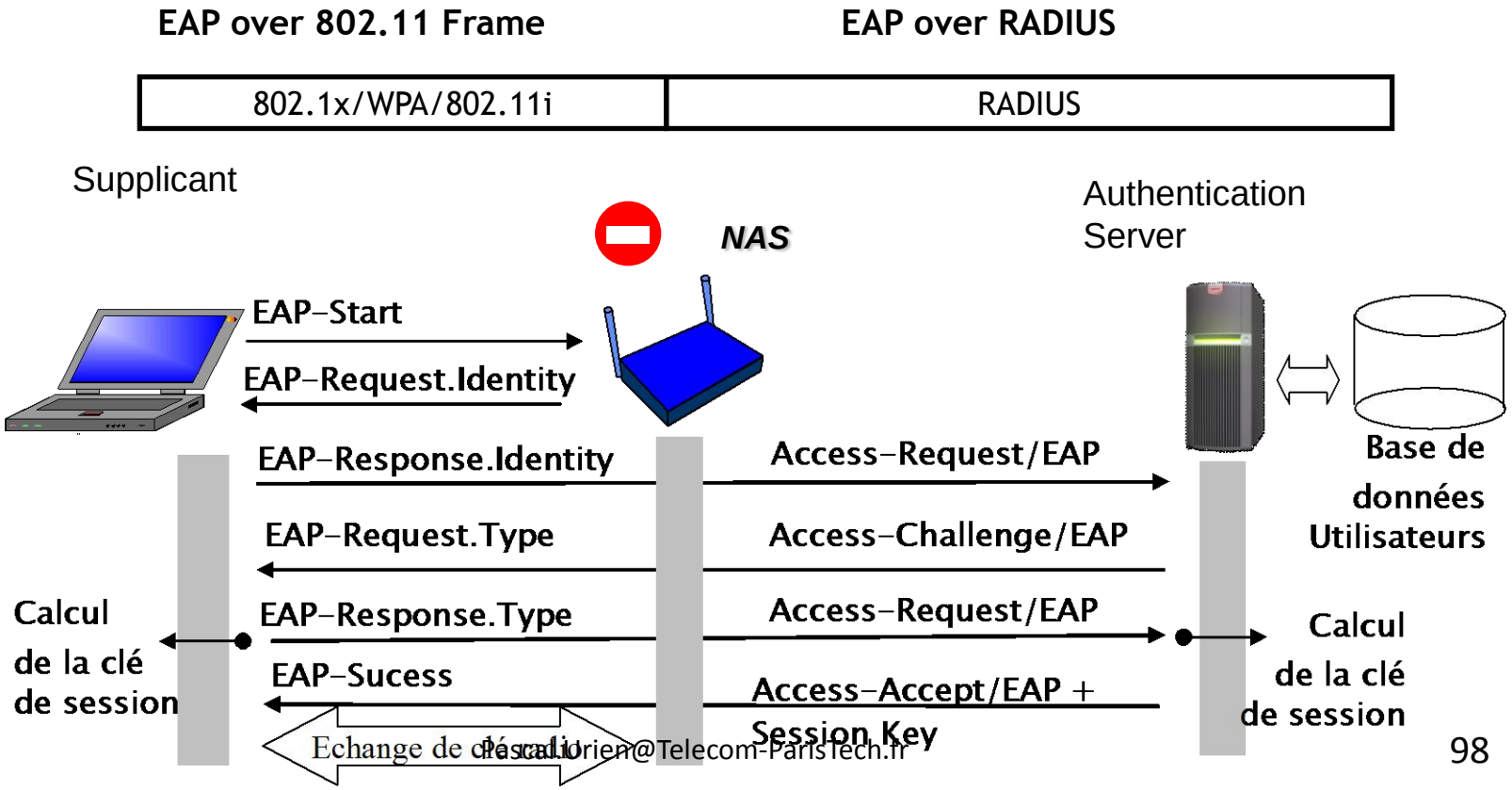
Le Passeport Electronique

- L'accès aux données du passeport est protégé par trois types de procédures
  - **BAC**, *Basic Access Control*. Une clé maître (kseed) est déduite du contenu MRZ. Deux clés sont calculées à partir de kseed et de deux valeurs aléatoires. Elles sont utilisées pour le chiffrement et l'intégrité des données échangées entre le passeport et le lecteur sans contact.
  - **AA**, *Active Authentication*. Une clé RSA privée stockée dans le passeport prouve l'authenticité du passeport (mesure anti-clonage).
  - **EAC**, *Extended Authentication Access*. Une procédure réalisant une mutuelle authentification (Diffie-Hellman sur courbe elliptique) entre le passeport et le lecteur sans contact..

Identité de Réseau

2001, IEEE 802.1X – WLAN 2/4

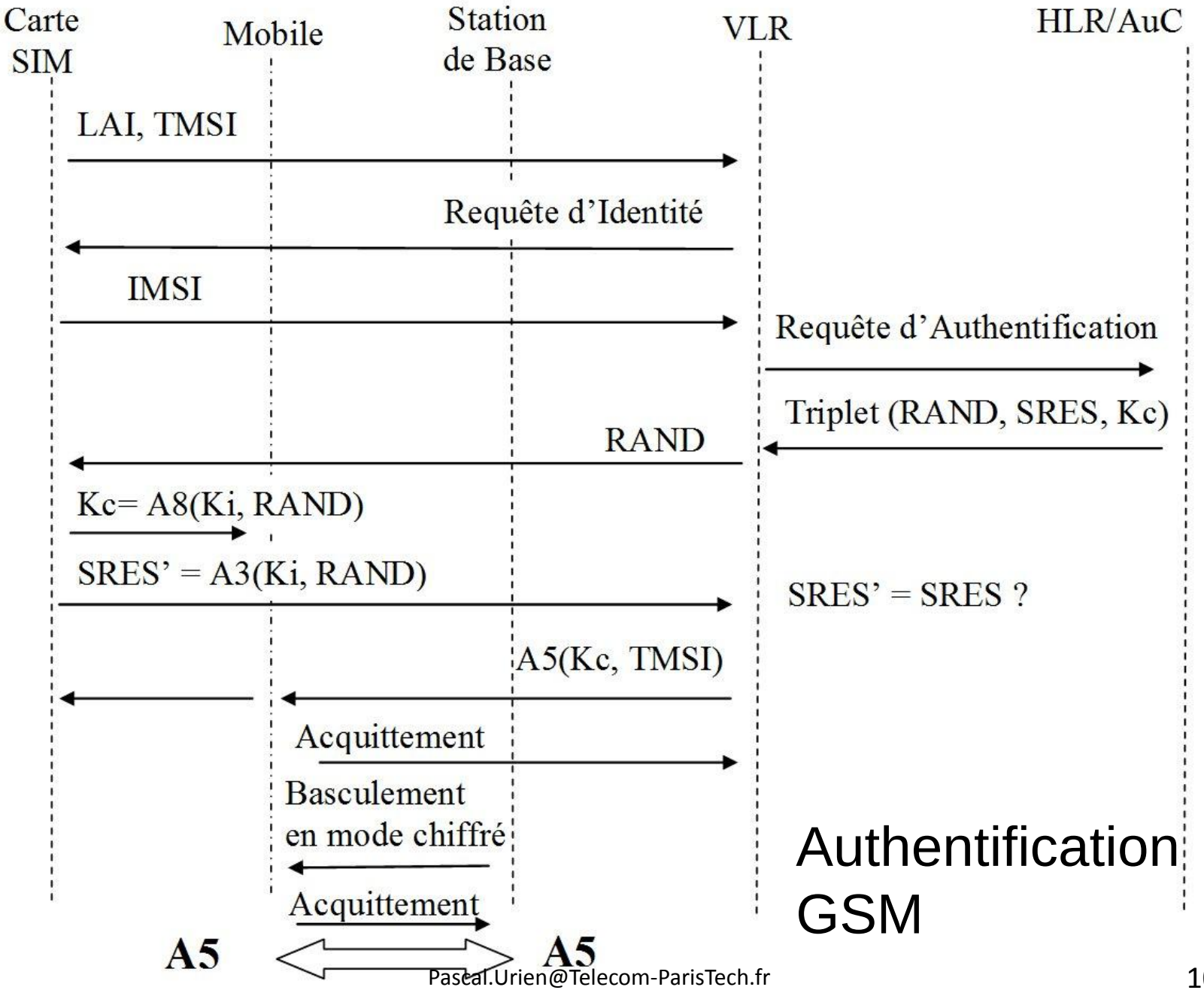
- Trois entités, Supplicant (client), Authenticator (Access Point), Authentication Server (serveur RADIUS).
- Authentification Mutuelle entre Supplicant et Authentication Server
- Echange de clé pour la sécurité radio



EAP

- Extensible Authentication Protocol
  - Un protocole parapluie qui transporte de multiples méthodes d'authentification
    - EAP-SIM, EAP-AKA importation de l'architecture d'authentification du GSM et UMTS
    - EAP-TLS, infrastructure d'authentification basée sur des clés asymétriques et des certificats
    - EAP-MSCHAPv2, authentification à base de mot de passe pour les plateformes Windows
    - Etc…
  - Triplet d'authentification (EAP-ID, Méthode d'authentification, lettre de crédit)
- L'authentification est la clé de voûte de la sécurité des réseaux sans fil IP émergents
  - Wi-Fi (IEEE 802.11), Wi-Max (IEEE 802.16), Wi-Mobile (IEEE 802.20), WirelessUSB (IEEE 802.15).

Carte SIM et USIM



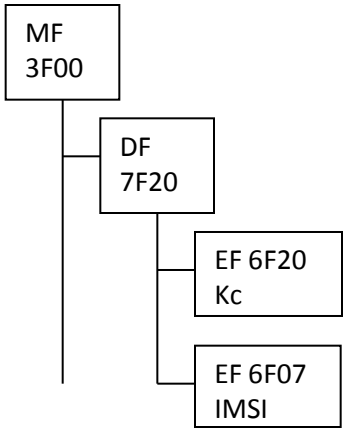
Authentification GSM

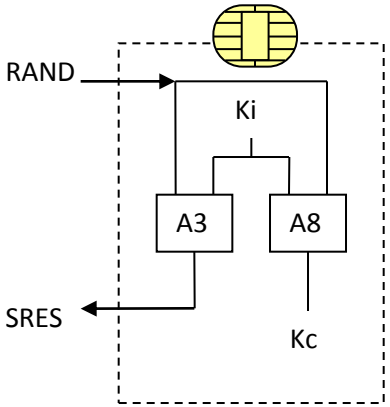
L'identité sur le GSM – GSM

- Une authentification à deux facteurs (PIN code + clé secrète Ki)
- Une carte SIM supporte 21 commandes réparties en quatre groupes
- Gestion des fichiers ISO 7816 (10)
  - Un répertoire Télécom contient des informations telles que les annuaires utilisateur.
  - Un répertoire GSM contient des fichiers tels que l'identité de l'abonné (IMSI)
- Contrôle des accès (5) aux fichiers (PIN codes, PUK, ..)
  - L'usage des fichiers et de l'algorithme d'authentification sont contrôlés par le PIN code (blocage après trois faux codes).
- Un algorithme d'authentification symétrique A3/A8, associé à une clé Ki de 128 bits (commande RUN_GSM_ALGORITHM).
- A partir d'une valeur d'entrée de 16 octets, il produit 12 octets, les 4 premiers constituent la signature SRES, et les 8 suivants la clé Kc (mais 10 bits sont forcés à zéro) utilisée pour le chiffrement des paquets voix (avec l'algorithme A5)
  - L'algorithme COMP128-1 a été craqué en 1998, en 219 vecteurs
  - L'algorithme A5/1, version forte, a été craqué en 99
  - L'algorithme A5/2, version faible, a été craqué en 99
- Transfert de données (5) avec le mobile (messages SMS,…)

La carte SIM

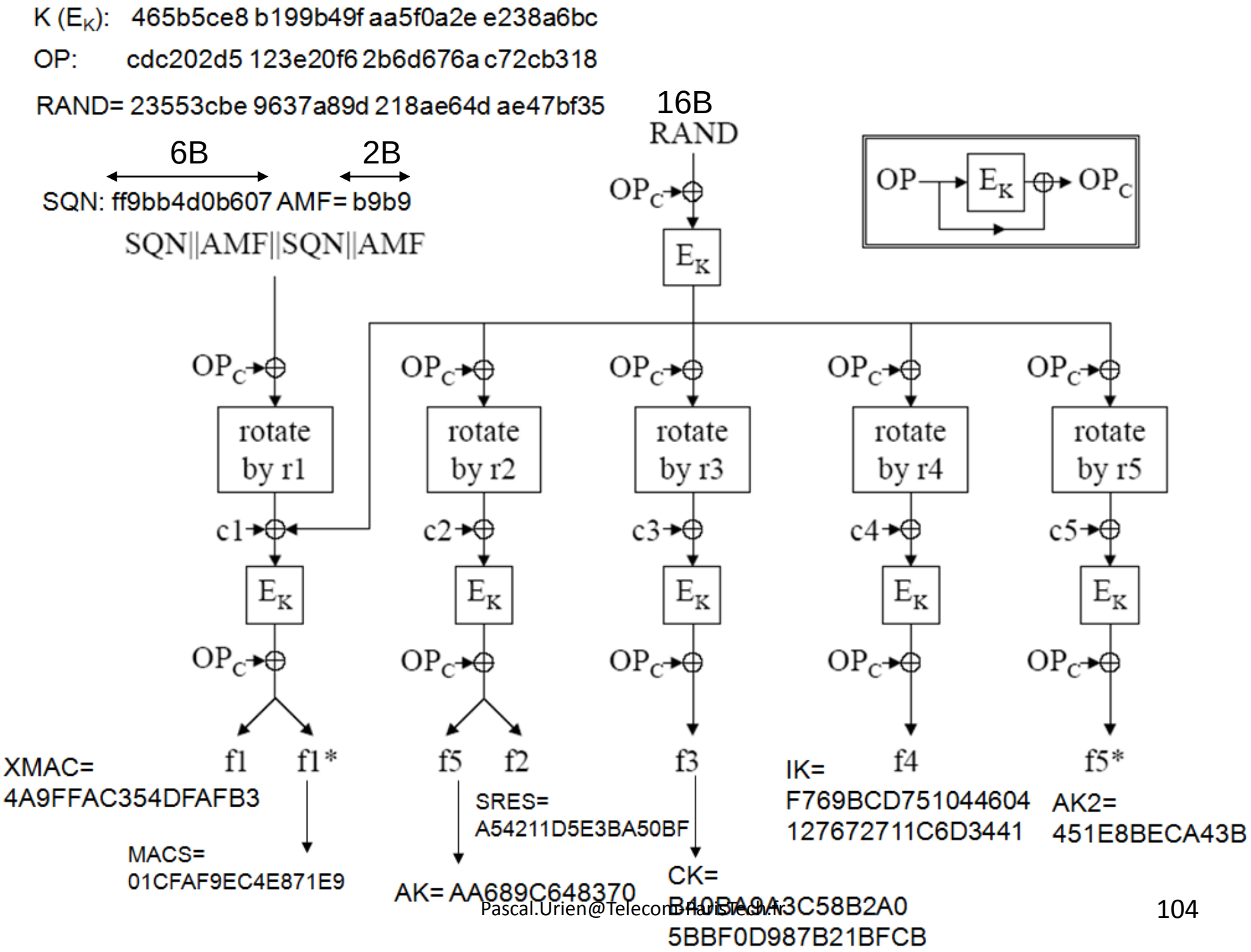
- L'information est organisée en répertoires et fichiers
- Quelques données
  - IMSI
  - Deux répertoires téléphoniques
  - Un fichier SMS
- Quelques procédures
  - RUN_GSM_ALGORITHM, calcule l'algorithme A3/A8





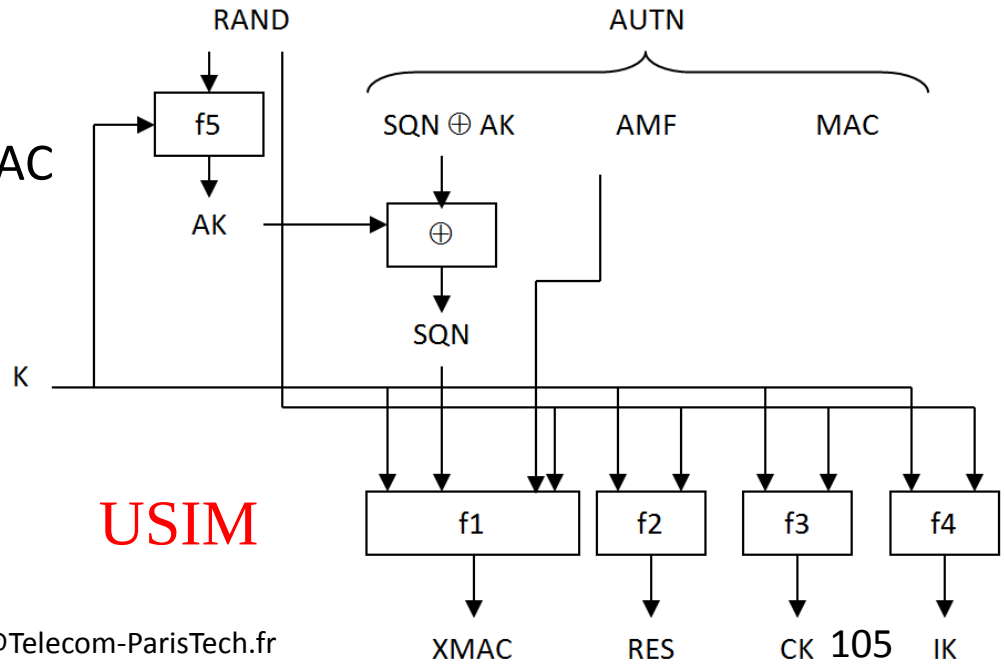
// Run_Gsm_Algorithm(RAND) **RAND**
>> A0 88 00 00 10 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF
<< 9F 0C
>> A0 C0 00 00 0C
<< FE 67 7C 9D B8 DD F1 B1 DE 27 18 00 90 00

SRES **KC**


K (E_K): 465b5ce8 b199b49f aa5f0a2e e238a6bc
OP: cdc202d5 123e20f6 2b6d676a c72cb318
RAND= 23553cbe 9637a89d 218ae64d ae47bf35
16B
RAND
6B
2B
SQN: ff9bb4d0b607 AMF= b9b9
SQN||AMF||SQN||AMF
OP
E_K
OP_C
rotate by r1
rotate by r2
rotate by r3
rotate by r4
rotate by r5
c1
c2
c3
c4
c5
f1
f1*
f5
f2
f3
f4
f5*
XMAC=
4A9FFAC354DFAFB3
MACS=
01CFAF9EC4E871E9
SRES=
A54211D5E3BA50BF
AK= AA689C648370
CK=
B40BA9A3C58B2A0
5BBF0D987B21BFCB
IK=
F769BCD751044604
127672711C6D3441
AK2=
451E8BECA43B

La carte USIM

- Le module UICC stocke au moins une application USIM
  - Le fichier EF_DIR contient la liste des applications USIM
- Au moins deux applications peuvent être activées simultanément (notion de canaux logiques
  - L'index de l'application est indiqué dans les deux derniers bits de l'octet CLA
- L'algorithme d'authentification AKA est réalisé par la commande AUTHENTICATE (INS=88) command
- Exemple
- >> 00 88 00 00 20 RAND || AUTN
- << DB 28 SRES || CK || IK 9000
  - AUTN:= SQN ⊕ AK || AMF || XMAC

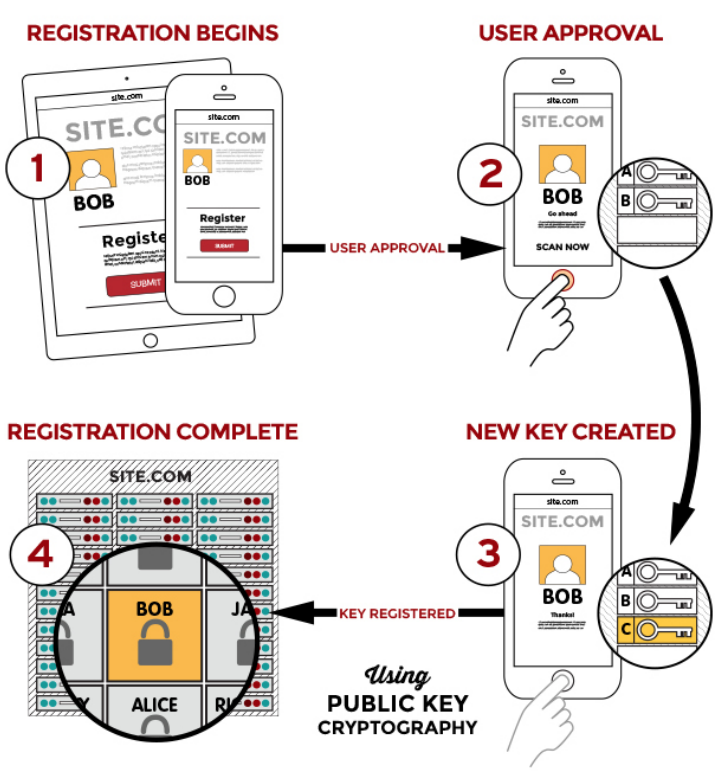


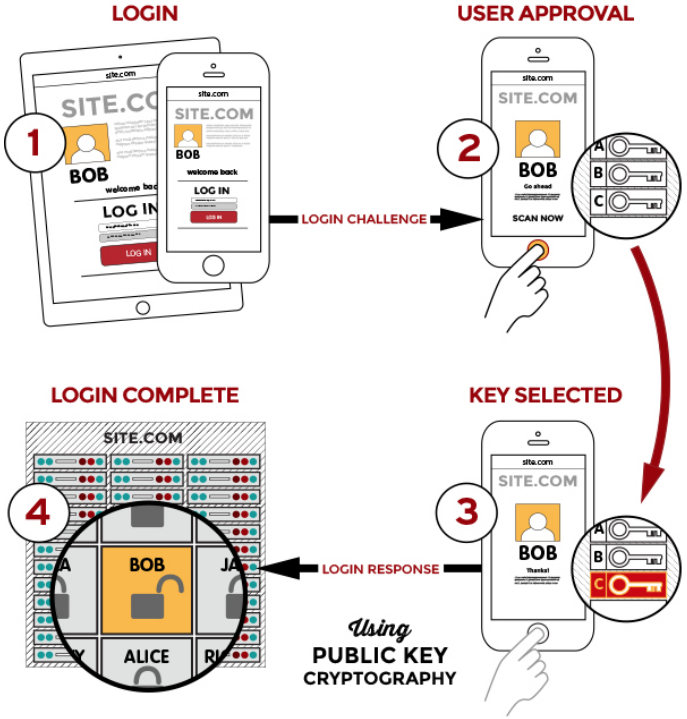
FIDO

FIDO

- FIDO est un système d'identité
- 3 procédures
  - Enregistrement
  - Authentification
    - Confirmation
  - De-Enregistrement
- Deux architectures
  - UAF, *Universal Authentication Framework* utilise la PKI et la biométrie (emprenite, voix, photo…) ou le TPM
  - U2F, *Universal 2nd Factor*, authentification à deux facteurs

FIDO en bref





UAF: une paire de clés asymétrique est crée après l'authentification biométrique de l'utilisateur

U2F: : une paire de clés asymétrique est crée après l'authentification (code PIN) l'utilisateur

Pendant la phase du login/authentification l'utilisateur prouve la possession d'une clé privée.

UAF Model

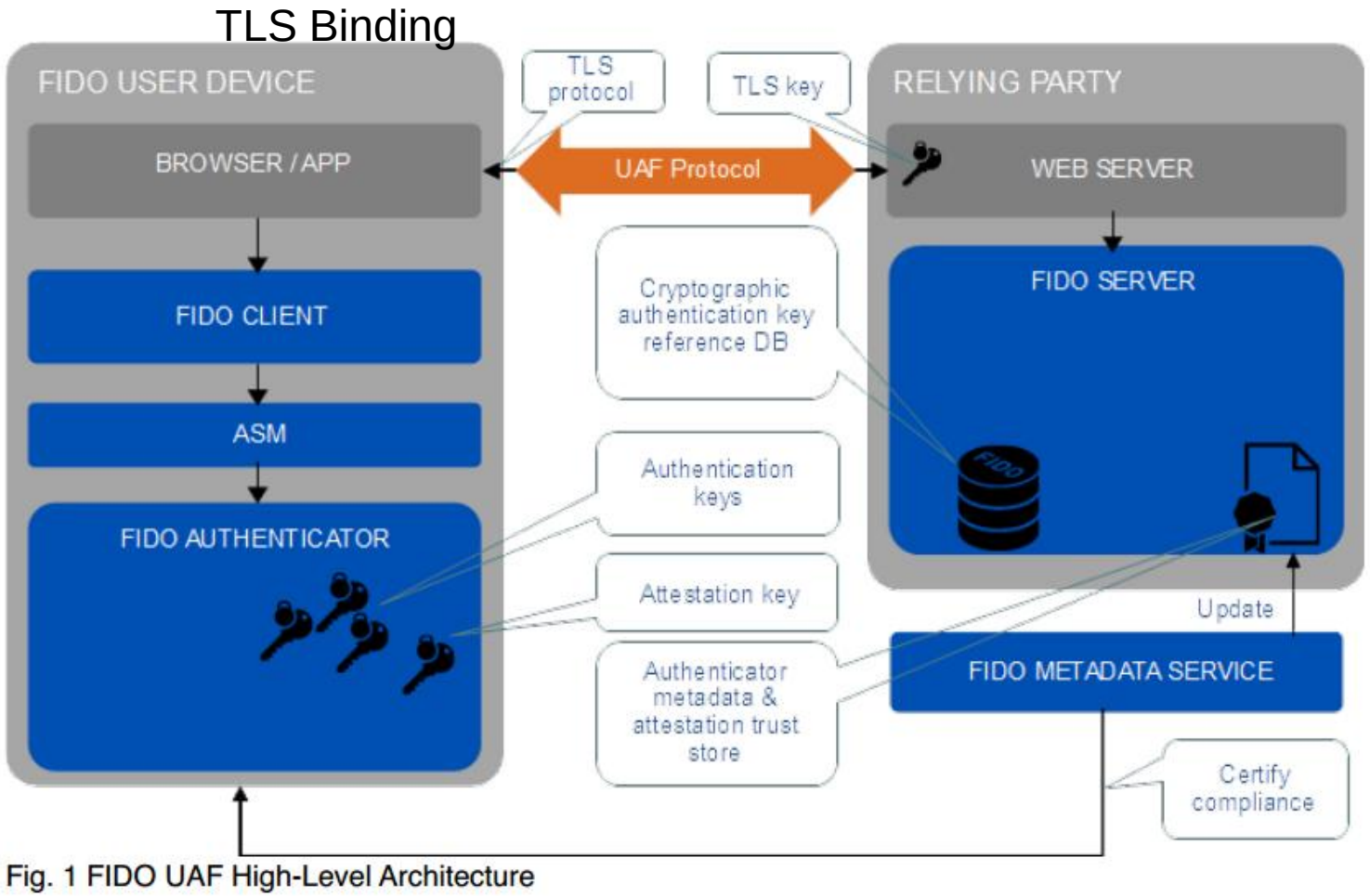


Fig. 1 FIDO UAF High-Level Architecture

FIDO UAF AUTHENTICATOR-SPECIFIC MODULE (ASM) API

*This document uses WebIDL to define UAF protocol messages. Implementations must serialize the UAF protocol messages for transmission using UTF-8 encoded JSON [RFC4627].

UAF Registration

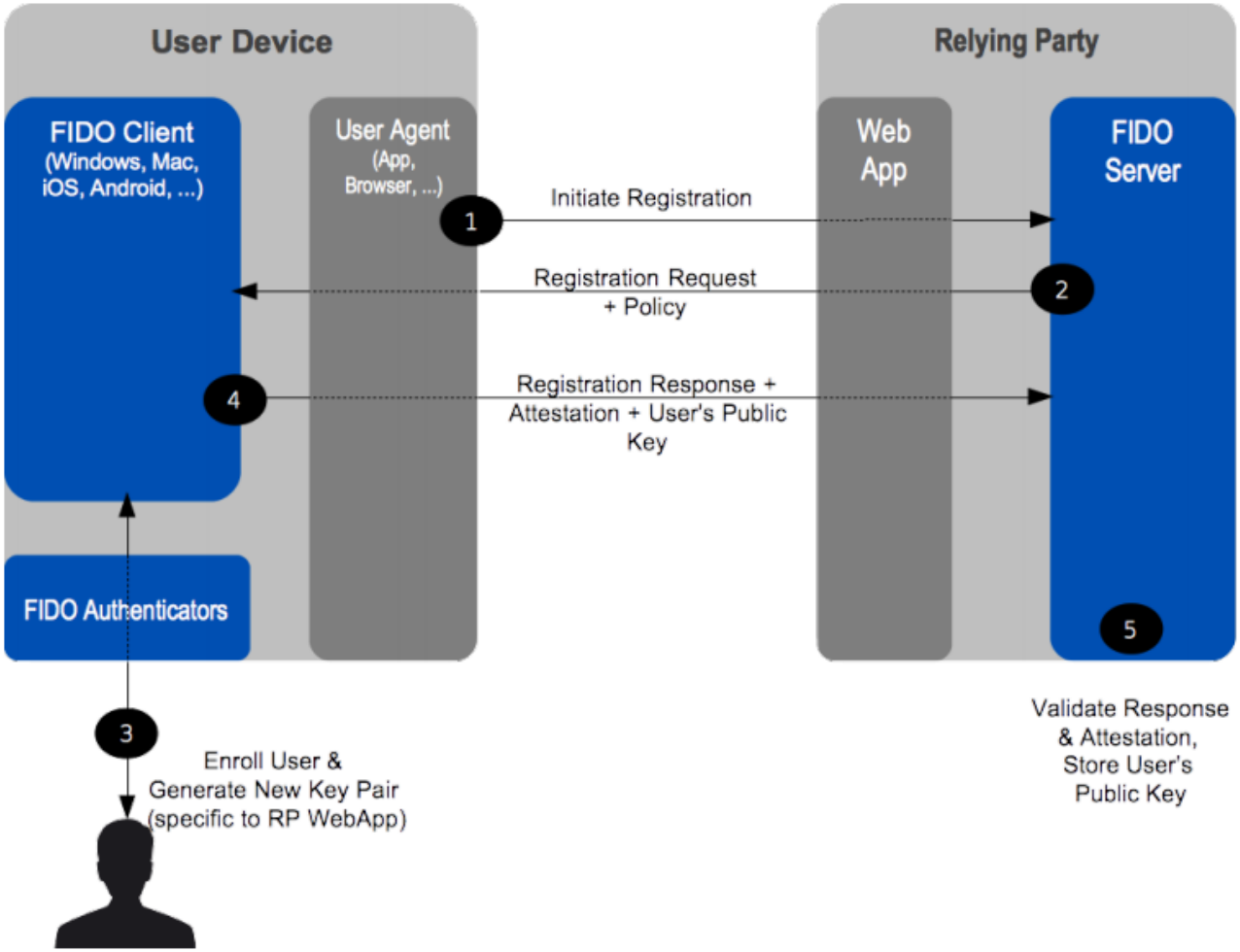


Fig. 2 Registration Message Flow

UAF Authentication

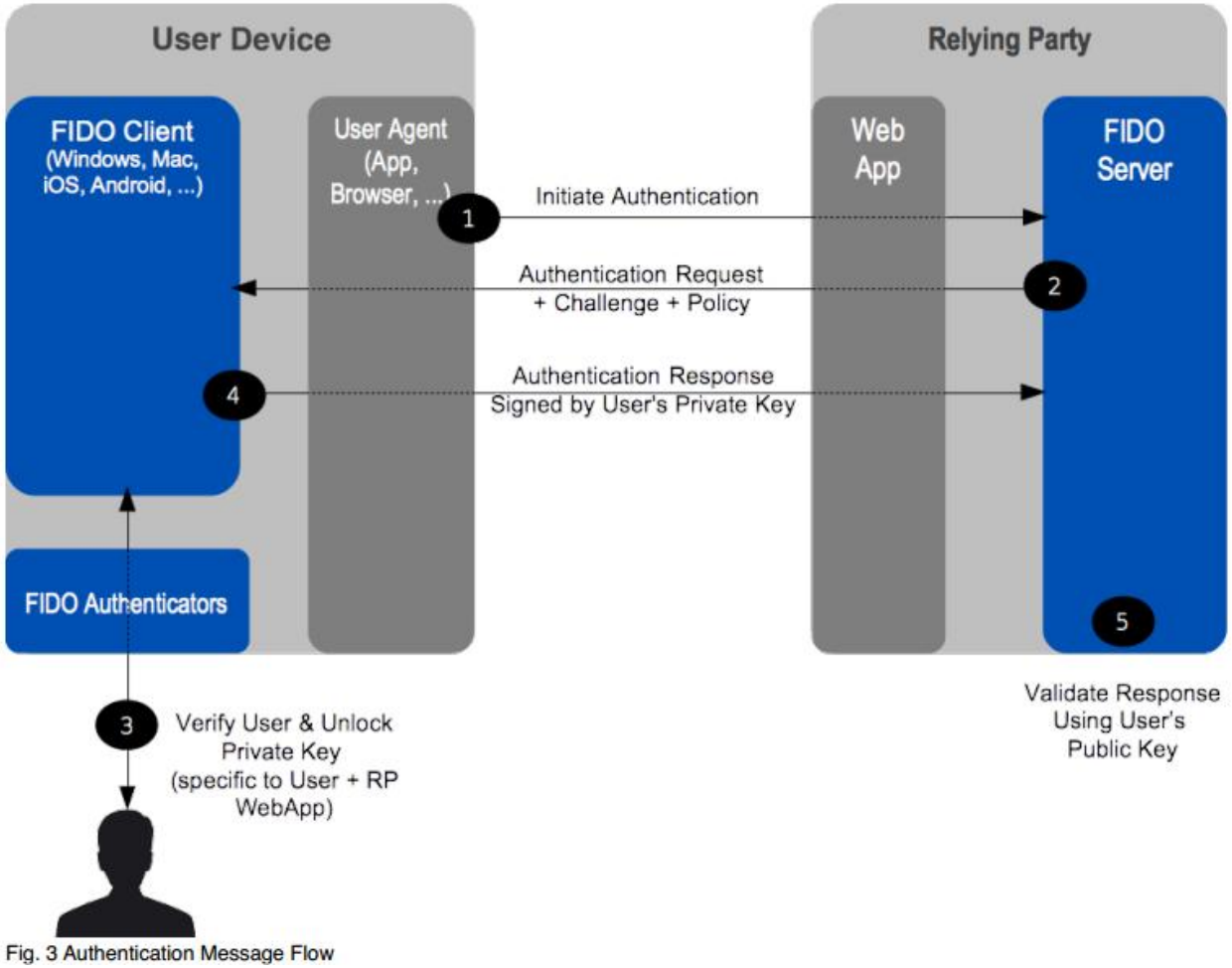


Fig. 3 Authentication Message Flow

UAF Confirmation

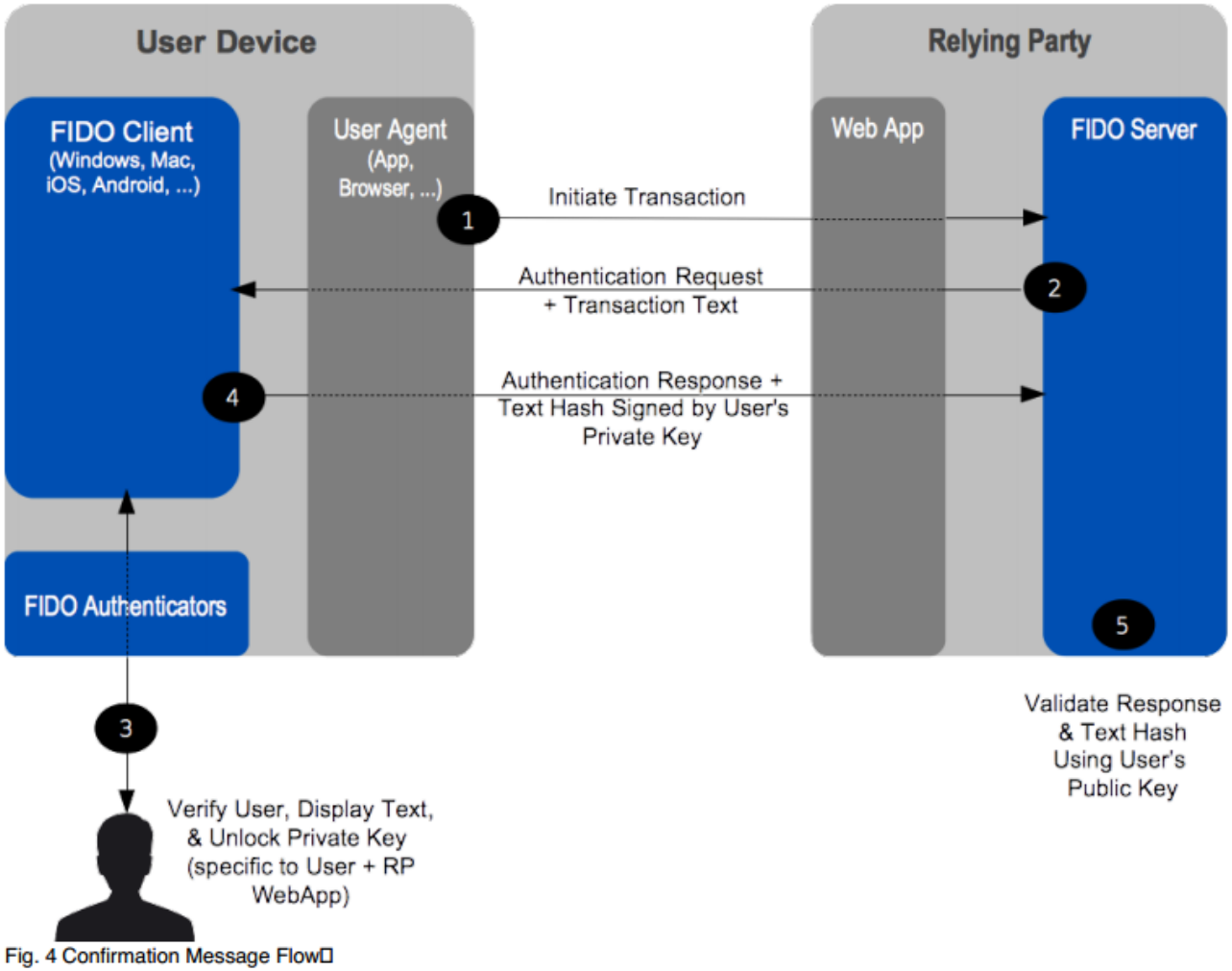


Fig. 4 Confirmation Message Flow

UAF De-Registration

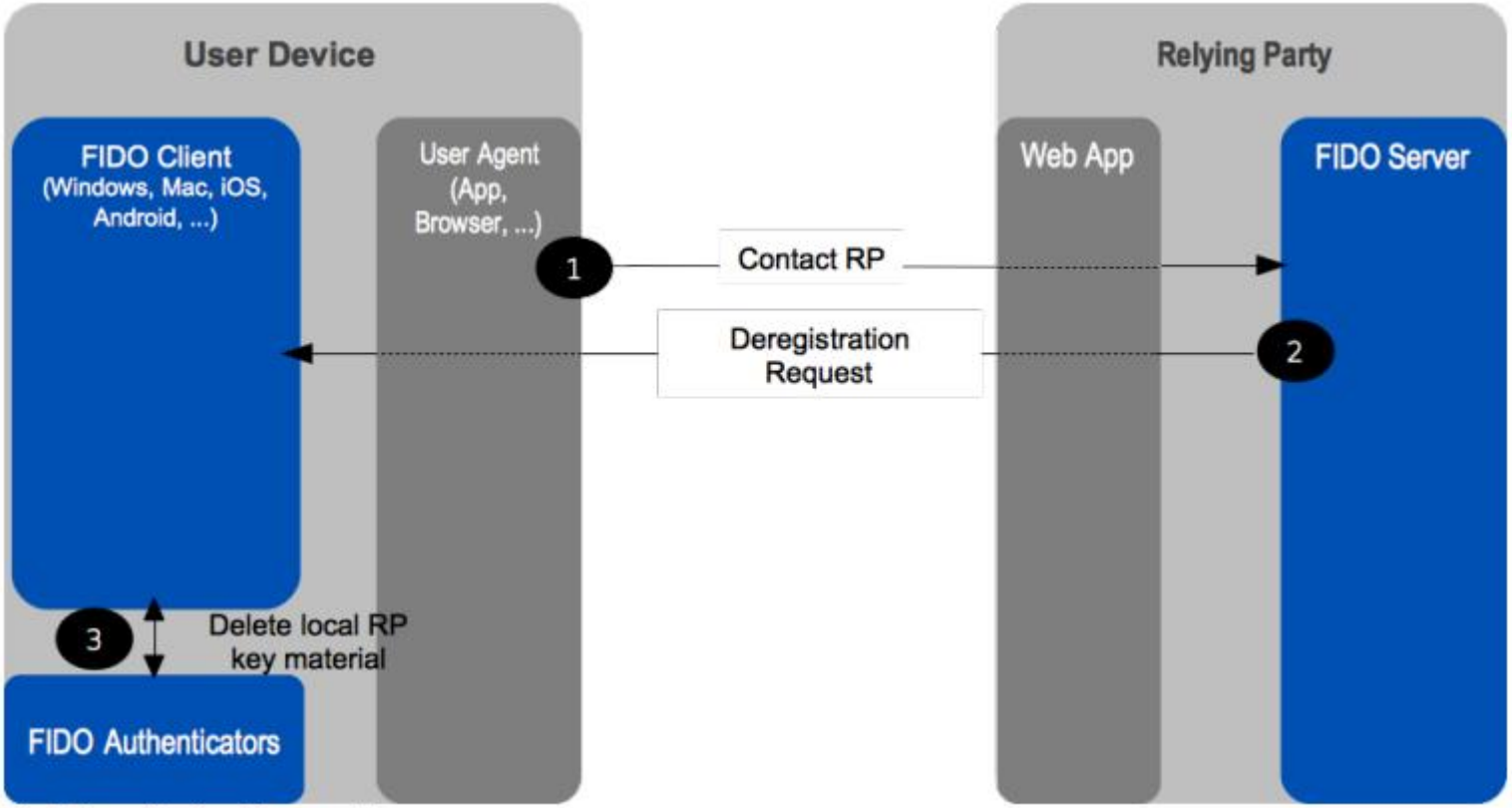


Fig. 5 Deregistration Message Flow

Universal 2nd Factor (U2F)

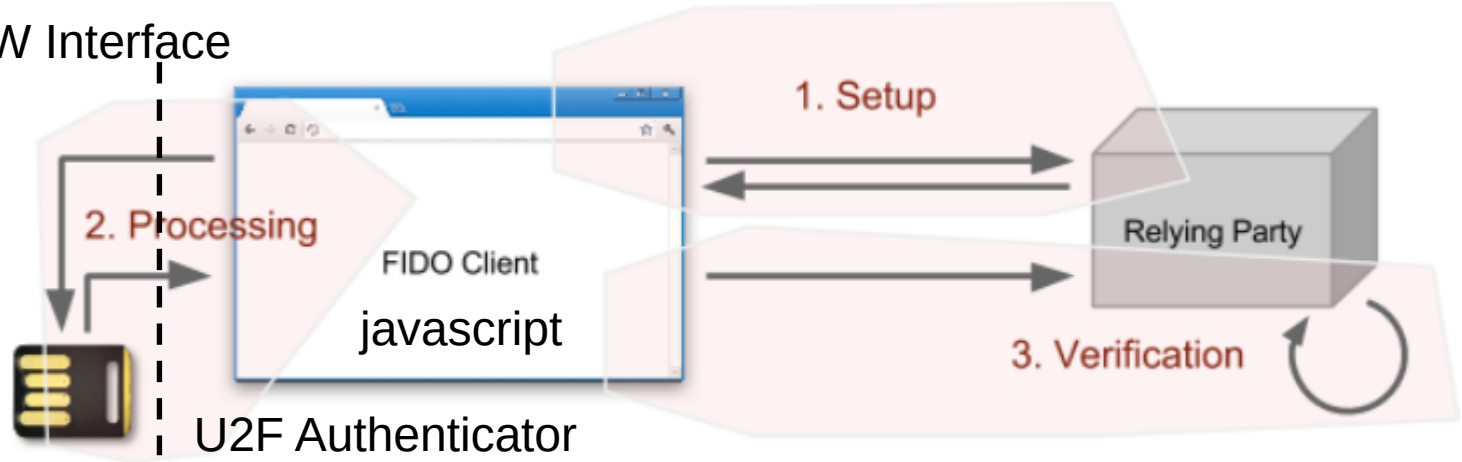
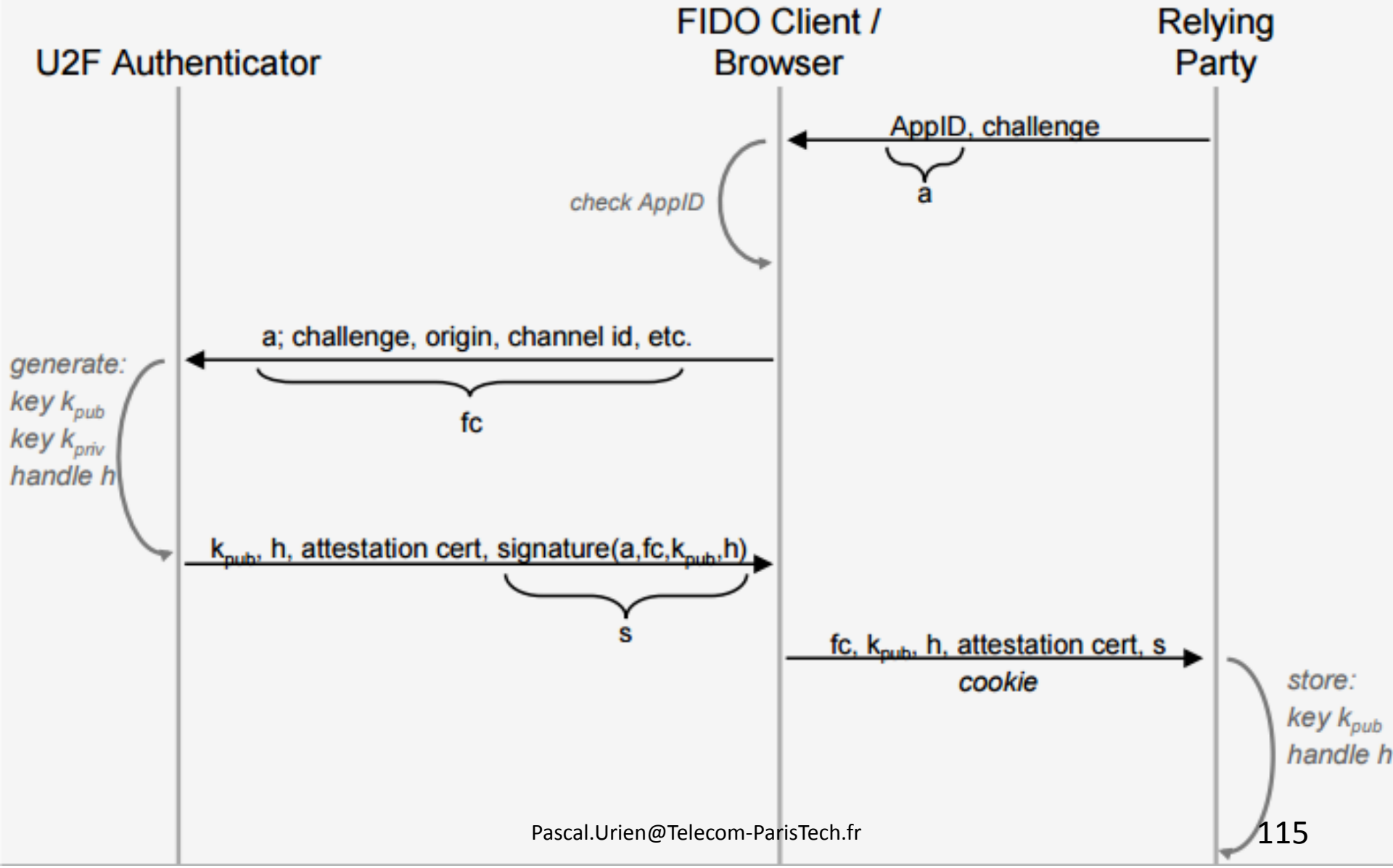


Fig. 1 Three phases of Registration and Authentication

- Utilisation de javascript coté client
- Plusieurs U2F Authenticators
  - Carte à puce
  - Token HID
  - Device Bluetooth

U2F Registration



U2F Authentication