Public Notary Transparency : des journaux publics en ajout seul et leurs usages avec TLS

Florian Maury (ANSSI)

11 juillet 2016

Rappel du contexte

600+ autorités de certification :

- nombreux rapports d'incident (émissions non-sollicitées ou certificats invalides)

Solutions possibles :

- **changer** de système de confiance ?
- **réduire** le nombre d'autorités de certification ?
- **renforcer** les exigences ?
- **détecter** les anomalies ?

Journaliser les certificats émis

Journalisation publique des certificats émis :

- base de données classique **inadaptée**
  - propriétés d'intégrité trop faibles

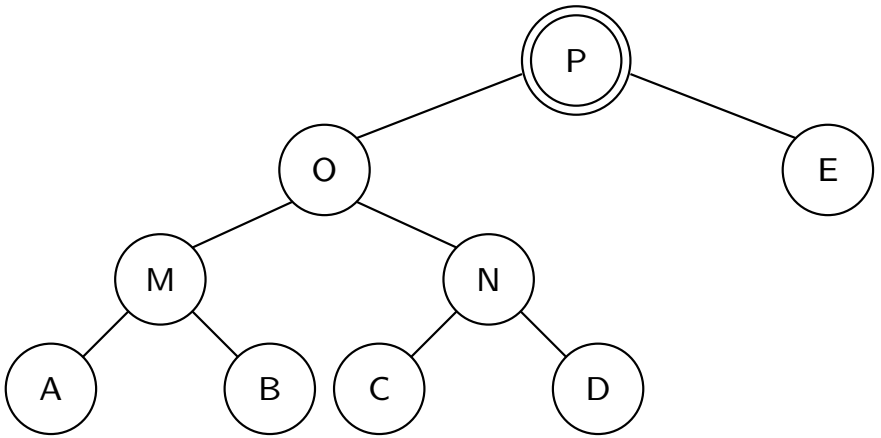
Propriétés attendues :

- **publique**
- **en ajout seul**
- **vérifications efficaces**

Structure d'un journal en ajout seul

Repose sur les **fonctions de hachage cryptographiques** (e.g. SHA-2) :

- arbre de Merkle (1979)



Exemples d'usage : Git, Bitcoin, Bitorrent

Certificate Transparency

Des journaux de certificats pour TLS

Principe général de Certificate Transparency (CT)

IETF :

- RFC 6962 (*experimental status*)
- WG *trans* : *draft* rfc-6962-bis-16

Objectifs :

- **journaliser les certificats** émis par les autorités de certification (AC) **publiques**
- permettre la **détection** des émissions défectueuses, frauduleuses ou indésirables
- ~~prévenir les interceptions TLS~~

Certificate Transparency (CT) :

le synoptique théorique

De nouveaux acteurs

Acteurs existants :

- client TLS (navigateur)
- serveur TLS (serveur web)
- autorité de certification (émetrice de certificats)

CT introduit de nouveaux acteurs :

- **mainteneur de journaux** (*log*)
- **moniteurs** (*monitor*)
- **auditeurs** (*auditor*)

Phase 1 : soumission des certificats aux journaux

En théorie, **tout le monde peut soumettre** un certificat valide

Étape généralement faite par l'autorité de certification dans le *workflow* d'émission de certificats

Soumission à **plusieurs journaux souhaitables**

Phase 1 (bis) : engagement de journalisation du certificat

Journalisation du certificat **effective après un délai** :

- **engagement de journalisation** (objet cryptographique) remis immédiatement au soumissionnaire
  - cet engagement peut constituer une **preuve de dysfonctionnement** du journal

Phase 2 : distribution des engagements de journalisation

Engagements de journalisation **distribués aux navigateurs** :

- **vérification** qu'il y a eu soumission du certificat observé pendant une transaction HTTPS

Ces engagements sont appelés des **Signed Certificate Timestamp (SCT)**.

Phase 3 : vérification synchrone par le client TLS

Si les engagements sont :

- absents
  ou
- émis dans le futur
  ou
- avec des signatures invalides

Alors :

- **erreur** (nature non spécifiée dans la RFC)

Phase 4 : vérification asynchrone par le client TLS

Vérification de l'**honnêteté du journal** :

- le navigateur $\Rightarrow$ rôle d'**auditeur**
- demande aux journaux de **preuve de la journalisation effective** du certificat dont il détient un **engagement de journalisation**
- partage de la preuve de journalisation par rumeur
  - participation à la protection contre les **attaques par partionnement de vue**

Phase X : surveillance des journaux

Phase continue, en parallèle des autres ou *a posteriori* :

- rôle de **moniteurs**
  - effectué par les **titulaires de noms de domaine**
  - **recherchent dans les journaux** de certificats invalides, frauduleux, ou émis par erreur

Nécessite le téléchargement des archives complètes des journaux !

Certificate Transparency (CT)

en pratique

Implémentation(s) cliente(s)

Seul client TLS compatible à ce jour : **Chrom(e|ium)**

Implémentation **partielle** :

- pas de vérification asynchrone
- pas de mécanisme de rumeur (mais des remontées UMA)
- stockage des engagements de journalisation dans le cache, non exportable

Politique CT appliquée par Chrom(e|ium)

Critères :

- nombre d'engagements de journalisation fonction de la durée de validité du certificat
  - **2 à 3 engagements minimum** requis pour les EV
- engagements provenant de journaux **de confiance**
- au moins un engagement émis par Google et un autre émis par un tiers

Politique CT appliquée par Chrom(e|ium)

Critères :

Site EV : 

Site DV/OV : 

- au moins un engagement émis par Google et un autre émis par un tiers

Politique CT appliquée par Chrom(e|ium)

Critères :

- nombre d'engagements de journalisation fonction de la durée de validité du certificat
  - **2 à 3 engagements minimum** requis pour les EV
- engagements provenant de journaux **de confiance**
- au moins un engagement émis par Google et un autre émis par un tiers

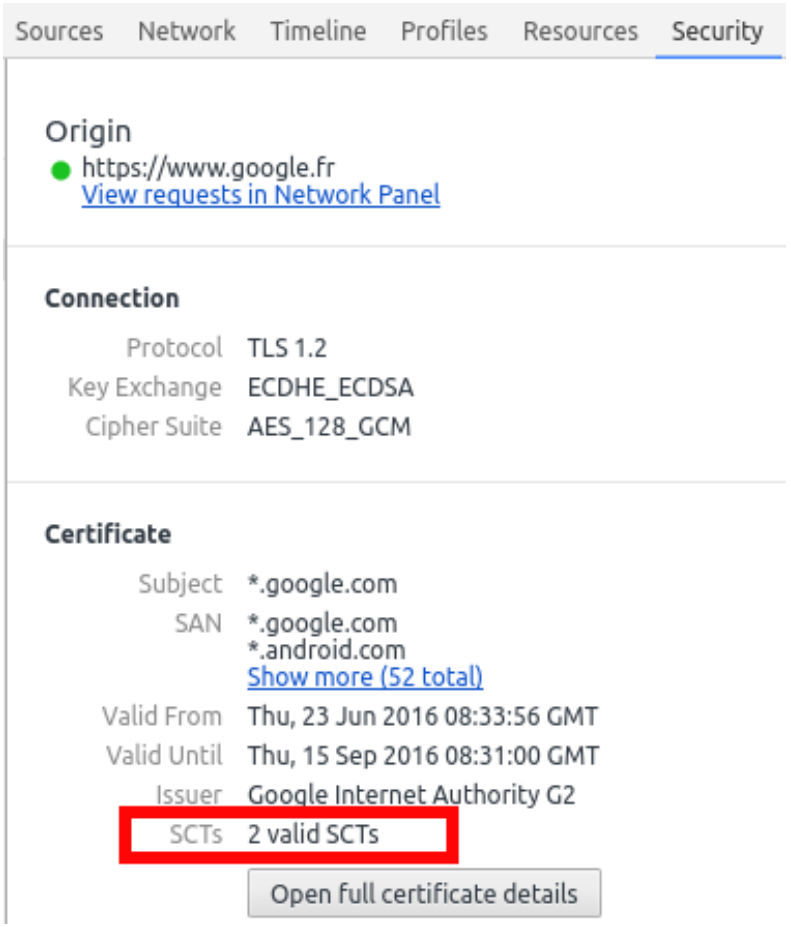
Impact de la politique CT de Chrom(e|ium)

Impact d'un échec de la validation synchrone :

- **perte du statut EV**
- pas d'impact pour les certificats DV et OV

Captures d'écran de CT dans Chromium

Captures d'écran de CT dans Chromium



Captures d'écran de CT dans Chromium

Les autorités de certification (AC)

Toutes les **AC dont le statut EV est reconnu par Chrom(e|ium)** participent à Certificate Transparency (CT)

Symantec, CNNIC, Let's Encrypt, StartCom/StartSSL et WoSign soumettent à des journaux tous leurs certificats (DV/OV/EV)

Les journaux

Les journaux, en chiffres :

- 9 journaux utilisés dont 3 administrés par Google

Taille du plus gros journal (Google Pilot) :

- 22M de certificats
- $\approx$ 40GB compressés avec GZip

Deux autres journaux de taille comparable. Le 4ième à $<$1M de certificats

Distribution des engagements

Distribution par le :

- le serveur web (extension TLS)
- le certificat (extension X.509)
- l'information de révocation du certificat (extension OCSP)

Distribution des engagements

Distribution par le :

- le serveur web (extension TLS) :
  - Nginx 1.9.0+, Apache et HaProxy (trunk)
  - exemple : google.fr, ritter.vg
  - Seul moyen ne requérant pas le concours de l'AC
- le certificat (extension X.509)
- l'information de révocation du certificat (extension OCSP)

Distribution des engagements

Distribution par le :

- le serveur web (extension TLS)
- le certificat (extension X.509) :
  - exemple : twitter.com, particuliers.societegenerale.fr
- l'information de révocation du certificat (extension OCSP)

Distribution des engagements

Distribution par le :

- le serveur web (extension TLS)
- le certificat (extension X.509)
- l'information de révocation du certificat (extension OCSP) :
  - exemple : sslanalyzer.comodoca.com
  - Let's Encrypt, à terme

Surveillance des journaux (« moniteurs »)

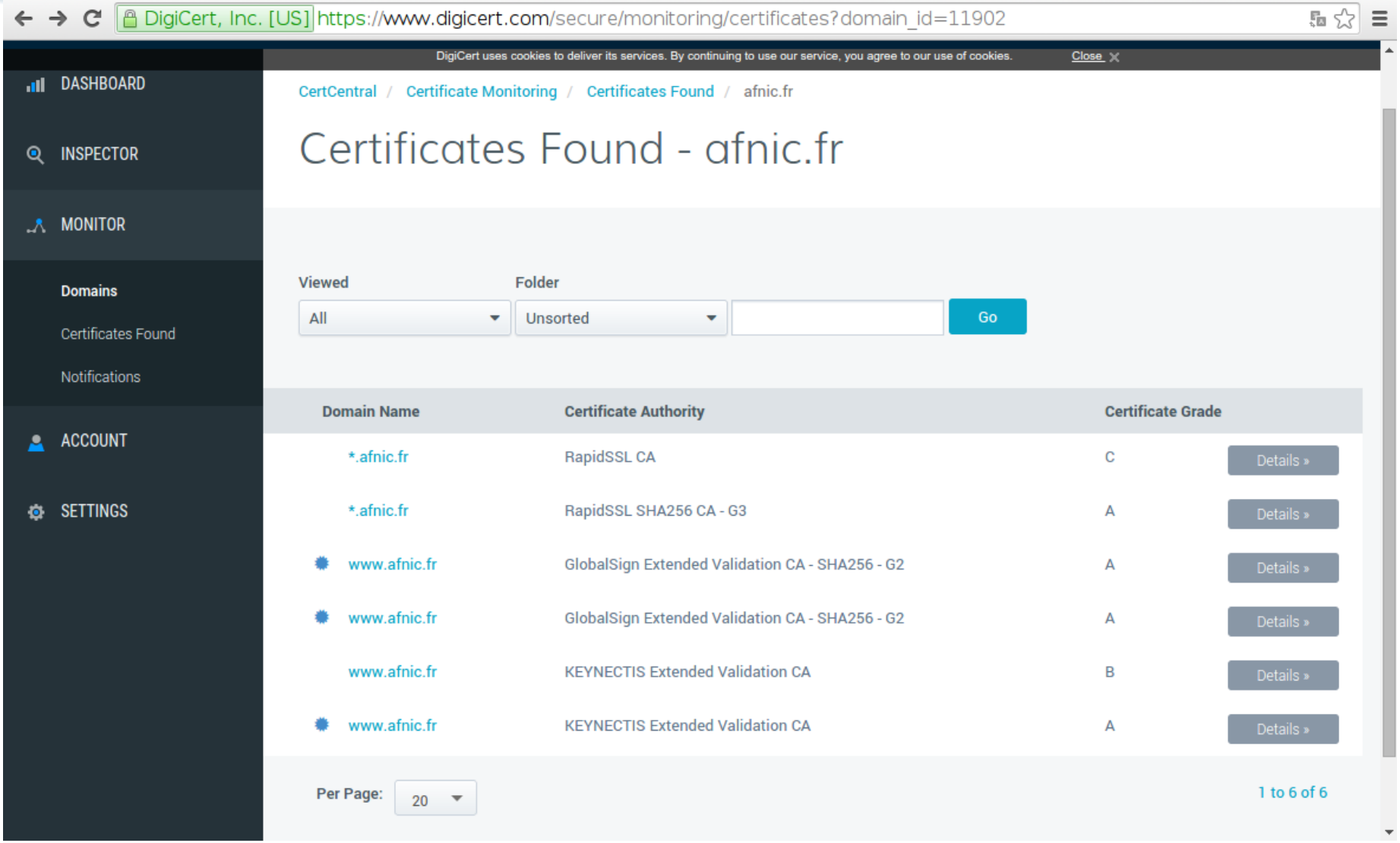
API en HTTP :

- permet l'interrogation par relais-proxy
- interface simple (GET/JSON)

Implémentation de référence par Google (Github)

Peu d'implémentations *open source* utilisables

Application web de Digicert : Certificate Monitoring



https://www.digicert.com/certcentral/certificate-monitoring.htm

https://crt.sh de Comodo

Comodo CA Ltd [GB] https://crt.sh/?id=5466060

crt.sh ID	5466060
Certificate Transparency	

Timestamp	Entry #	Log	Operator	URL
2014-11-03 12:01:31 GMT	5463892	Google 'Pilot' log	Google	https://ct.googleapis.com/pilot
2014-11-03 12:23:16 GMT	4717760	Google 'Aviator' log	Google	https://ct.googleapis.com/aviator
2014-11-04 01:43:54 GMT	2796523	Google 'Rocketeer' log	Google	https://ct.googleapis.com/rocketeer

SHA-256(Certificate)	5EACCA31C11510915204F887C8746C87DFD08EFFB5D2F6D635F49DB7EBEC366B
SHA-1(Certificate)	C8D42F592E53A6DBDAA297CBA983B35F8414ECAC
Certificate | ASN.1 **Run cablint** **Run x509lint**	Certificate: Data: Version: 3 (0x2) Serial Number: 36:bd:62:88:82:cc:f6:a3:1d:24:60:76:64:ff:25:e2 Signature Algorithm: sha256WithRSAEncryption Issuer: commonName = COMODO RSA Domain Validation Secure Server CA organizationName = COMODO CA Limited localityName = Salford stateOrProvinceName = Greater Manchester countryName = GB Validity Not Before: Nov 1 00:00:00 2014 GMT Not After : Oct 31 23:59:59 2019 GMT

Contremesures au partionnement de vue

This page left intentionally blank

Contremesures au partionnement de vue

Protocole encore non spécifié

- draft-ietf-trans-gossip-02

Création de miroirs par Google

Success stories :

Certificats détectés

avec Certificate Transparency

Incident Symantec

Symantec a émis :

- depuis 1995, environ **100000 certificats non-sollicités** « à des fins de test qualité »
- en 2015, un rapport initial **sous-estimant** le nombre de certificats émis en réalité

Google trouve, grâce à CT, la **preuve d'un certificat EV émis et non listé** dans le rapport

Incident Facebook (FB)

Deux certificats pour des sites satellites :

- émis par Let's Encrypt à la demande d'un prestataire
- **violant les politiques internes** de FB
- **non frauduleux, mais non sollicités** par FB
- détectés grâce à Certificate Transparency
- révoqués, suite à l'incident

Facebook vise à proposer au public son outil de monitoring

Conclusion

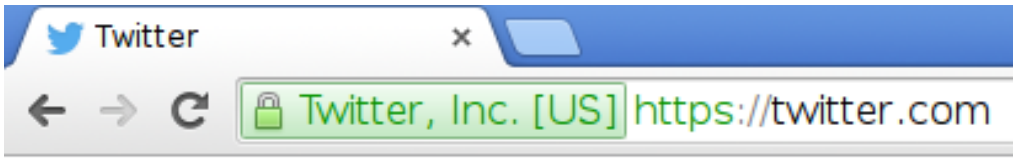
Conclusion : point d'attention №1

Certificate Tranparency (CT) :

- est **déjà déployé**
- contient plusieurs millions de chaines de certificats
- est **appliqué** par Chrom(e|ium)

Avec Chrom(e|ium) :

- un indicateur EV visible == Certificate Transparency OK



Conclusion : point d'attention №2

Que retenir de cette présentation ?

- utilisateurs :
  - les certificats EV apportent des **assurances supplémentaires**
- titulaires de noms de domaine :
  - **surveillez les registres CT !**

Conclusion : point d'attention №2

Que retenir de cette présentation ?

- développeurs :
  - il existe un besoin d'outils d'audit accessibles
- spécificateurs :
  - la version 2 du protocole est en cours de finalisation
  - le mécanisme de rumeur est nécessaire et reste à affiner

Perspectives : General Transparency

Constructions avancées avec des arbres de Merkle

Utilisation de deux journaux et d'un arbre de Merkle creux [1]

Exemples d'usage :

- CRL
- publication de politiques (e.g. MTA-STS)
- remplacement du DNS

1. *Sparse Merkle Tree*

Merci pour votre attention

Bibliographie

- Implémentation de référence de CT
- Rapport d'incident Symantec
- Rapport d'incident Facebook
- Site « Certificate Monitoring » de Digicert
- Site moniteur de Comodo
- Site implémentant un début de protocole de rumeur
- Liste des miroirs maintenus par Google
- General Transparency

General Transparency

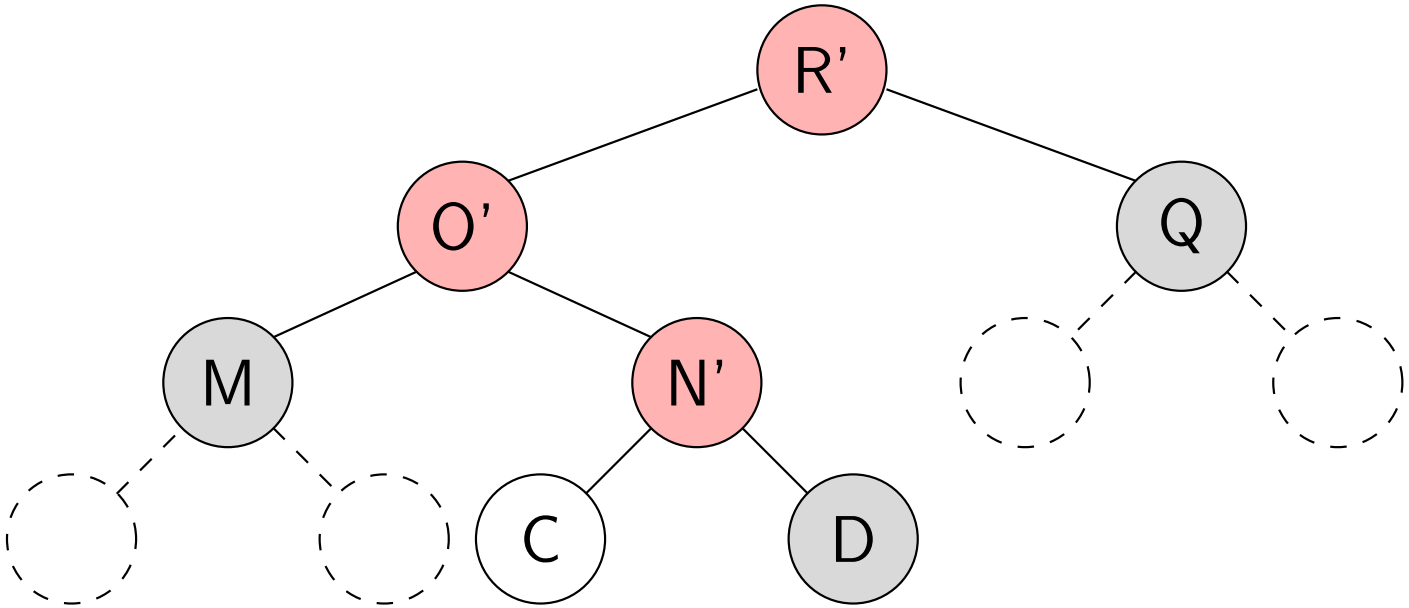
Constructions avancées avec des arbres de Merkle

Tables de hachage creuses et vérifiables :

- utilisation de deux registres et d'un arbre de Merkle creux
- interrogation en temps constant
- réponse de taille constante
- probabilité de collision négligeable (2^{-256})
- intégrité vérifiable de manière efficace

Cas d'usages : CRL, publication de politiques (MTA-STS), remplacement du DNS...

Preuve d'insertion demandée par un auditeur



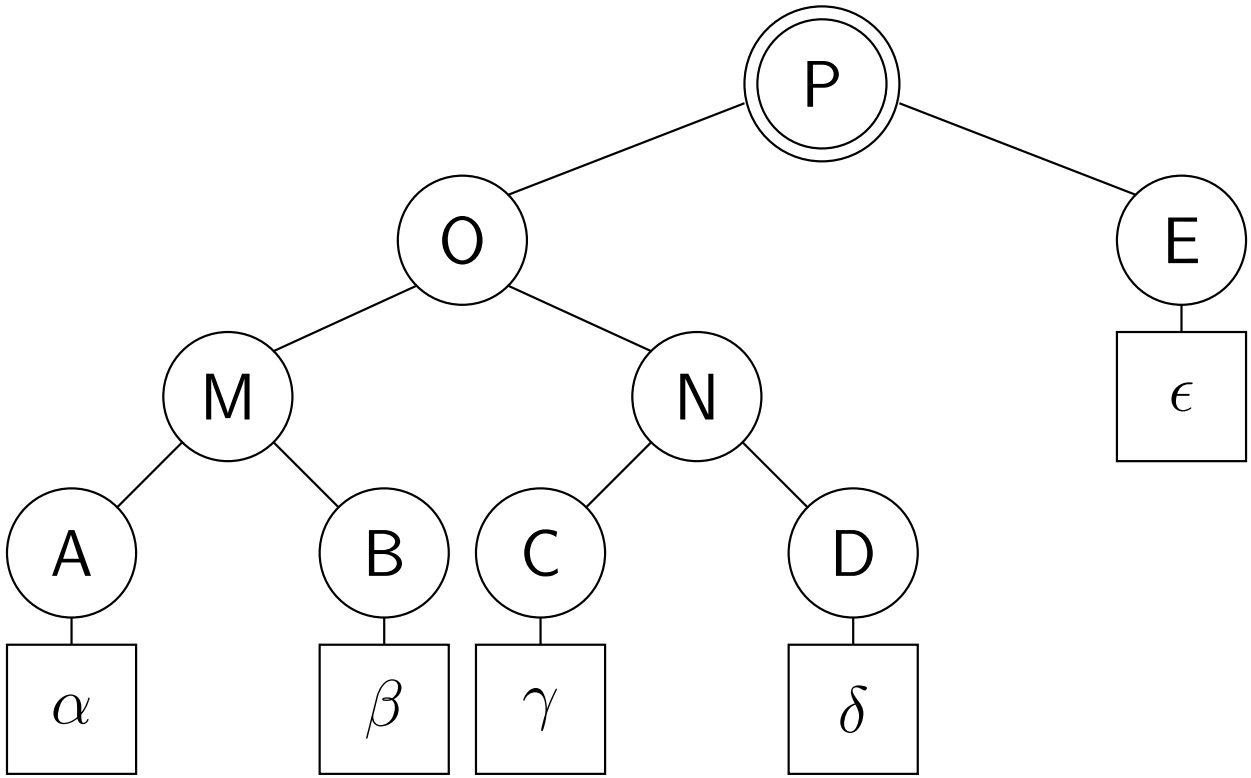
Preuve de présence de C. Les nœuds gris forment la preuve. Les nœuds roses sont calculés. `R'` doit être égal au `R` du précédent schéma.

Propriétés des arbres de Merkle

Propriétés :

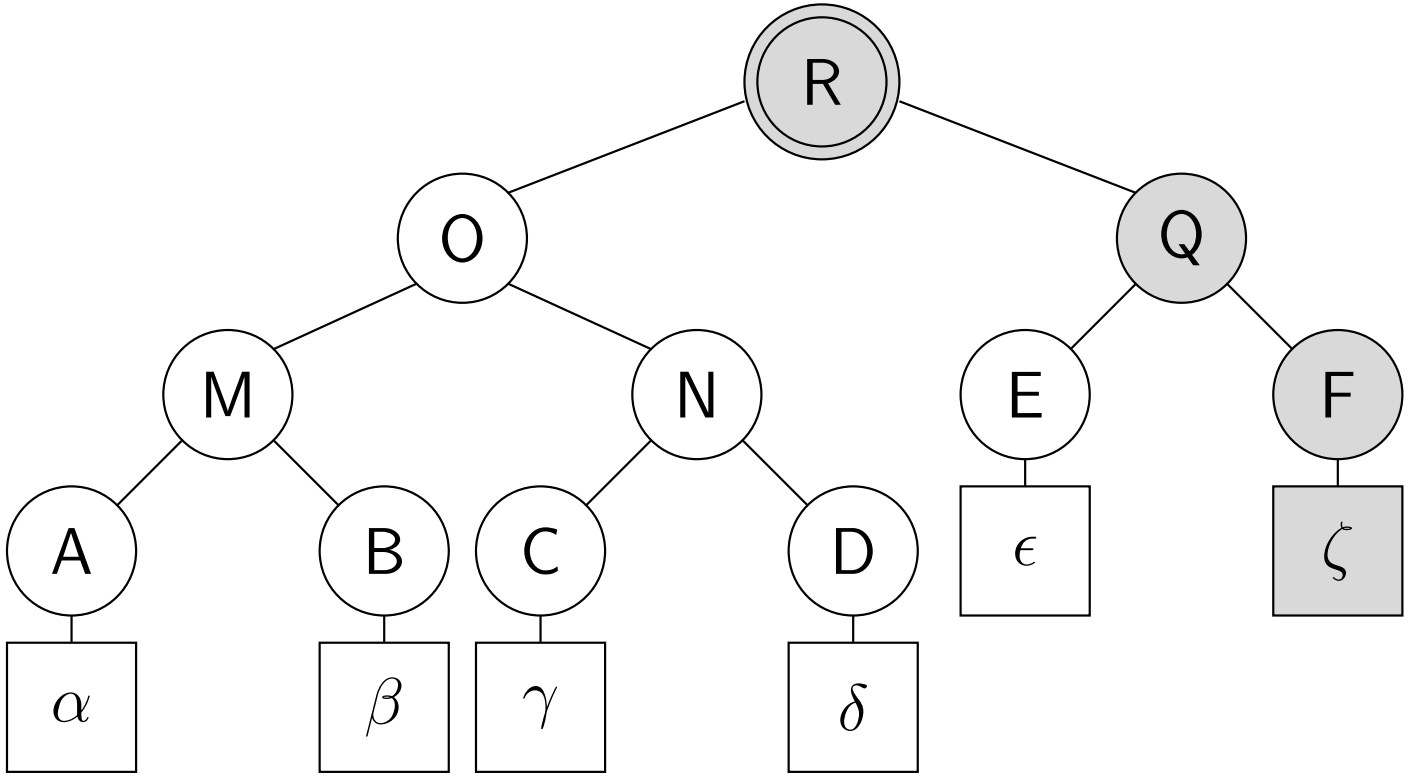
- croissance par **supplantation de la racine**
- **intégrité**
  - signature de la racine == signature de l'arbre complet
  - vérification efficace
- **auditabilité**
  - preuves de cohérence entre versions

Propriétés des arbres de Merkle



- preuves de cohérence entre versions

Propriétés des arbres de Merkle



Ajout de ζ à l'arbre. Entraine la création de `Q`, qui avec `O` forme une nouvelle racine `R`.

Propriétés des arbres de Merkle

Propriétés :

- croissance par **supplantation de la racine**
- **intégrité**
  - signature de la racine == signature de l'arbre complet
  - vérification efficace
- **auditabilité**
  - preuves de cohérence entre versions