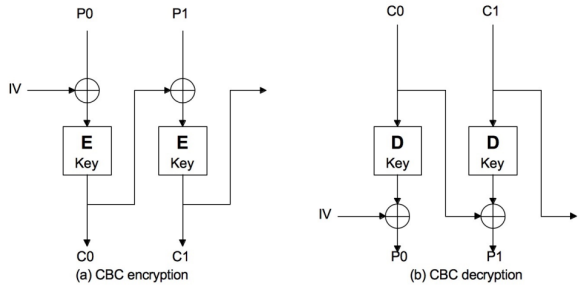
1 TPM

1.1 Chiffrements

1.1.1 symétrique

- Une seul clé pour crypter et décrypter
- Codage par bloc ou par bloc chainé



- `openssl enc -aes-256-cbc -e -in t.txt -out t.enc #encrypt`

  `openssl enc -aes-256-cbc -d -in t.enc -out t.txt #decrypt`

1.1.2 asymétrique

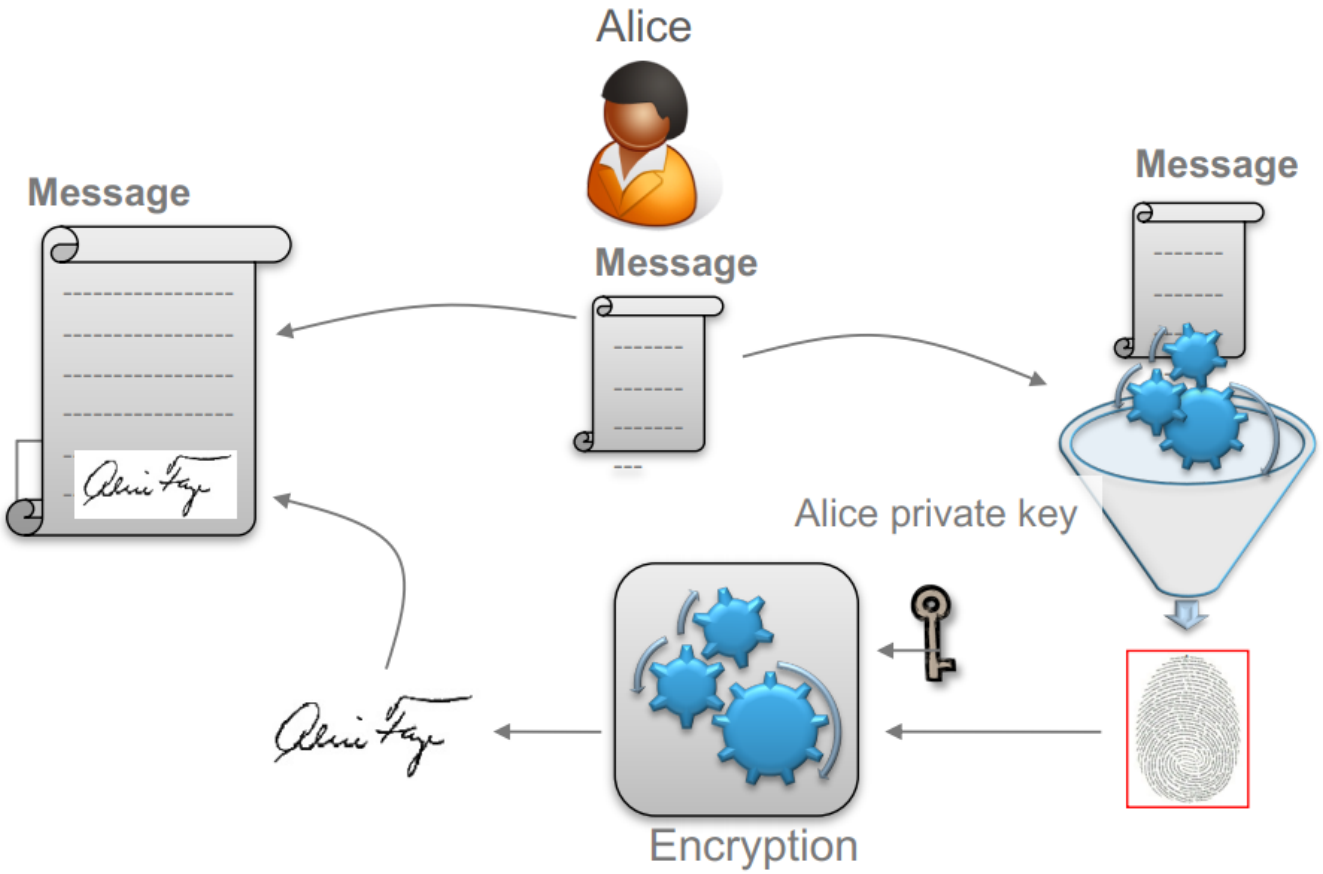
- Deux clés (publique et privée) clé publique disponible par des certificats (CMD pgp)
- Encrypt public - Decrypt private = confidentialité
- Encrypt private- Decrypt public = signature digitale

1.1.3 hash

Transforme un texte, document en un nombre de N bits unique (SHA-2, SHA-3, Blake2).
`md5sum file => a6a0e8d0522...` où `openssl dgst -md5 file`

1.1.4 signature

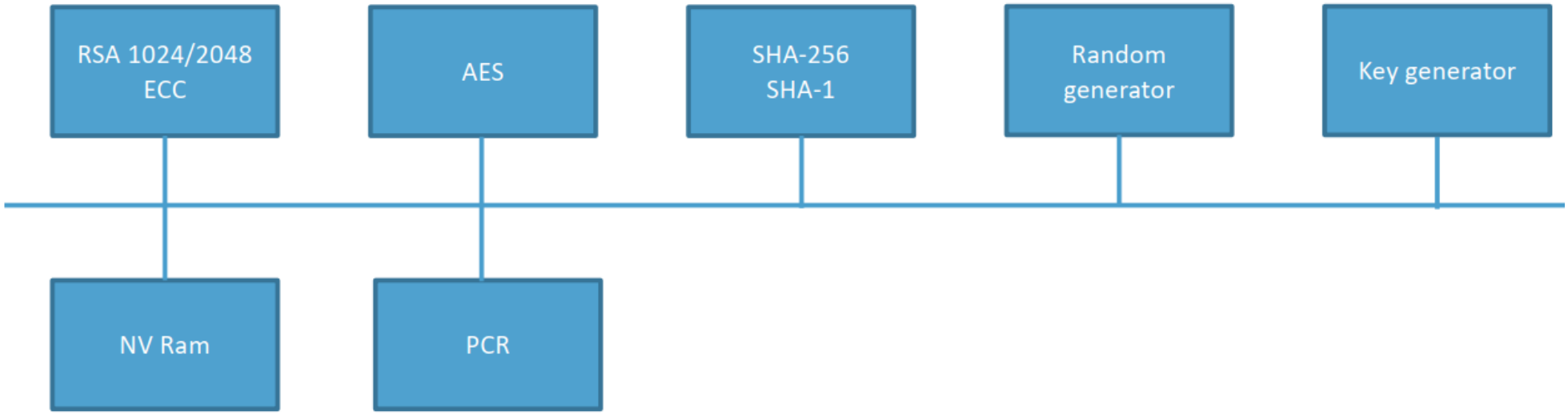
En deux parties: 1. Calcul du HASH puis encryptage avec clé privée.



1.2 Implémentations TPM

- discrete : Circuit dédié
- integrated : Partie du μC qui gère le TPM
- Hypervisor : virtuel fournis par personne fiable
- Software : virtuel pour faire des test pas sécurisé

1.3 Architecture interne



- RSA 1024/2048, ECC: Asymmetric algorithms, encrypt-decrypt, sign
- AES: Symmetric algorithm, encrypt-decrypt, sign
- SHA-256, SHA-1: hash function
- Random generator: create random value
- Key generator: Create key for asymmetric algo
- NV Ram: Store different objects (keys, data, ...) in NV Ram
- PCR (Platform Configuration Registers) stores hash values of different parts: code, files, partitions, ...

1.4 Hiérarchies

- endorsement : réservé au fabricant du TPM et fixé lors de la fabrication.
- platform : réservé au fabricant de l'hôte et peut être modifier par l'équipementier.
- owner : hiérarchie dédiée à l'utilisateur primaire du TPM peut être modifié en tout temps.
- null : réservé aux clés temporaires (RAM s'efface à chaque redémarrage)

1.5 Créer, utiliser clés

Create RSA endorsement key: `tpm2_createprimary -C e -G rsa2048 -c e_primary.ctx`

Create RSA platform key: `tpm2_createprimary -C p -G rsa2048 -c p_primary.ctx`

Create RSA owner key: `tpm2_createprimary -C o -G rsa2048 -c o_primary.ctx`

Create RSA null key: `tpm2_createprimary -C n -G rsa2048 -c n_primary.ctx`

1.6 Commandes principales

```
tpm2_createprimary -C o -G rsa2048 -c o_prim #créer un clé primaire owner
tpm2_getcap handles-transient #voir clé dans la RAM
tpm2_getcap handles-persistent #voir clé dans la NV-RAM
tpm2_evictcontrol -c o_primary.ctx # sauver une clé en NV-RAM
tpm2_flushcontext! -t ##effacer toute la RAM
tpm2_create -C o_prim -G rsa2048 -u child_pub -r child_priv #créer clé enfant
tpm2_load -C o_prim -u child_pub -r child_priv -c child #charger clé enfant
shred passwd, rm -f passwd #supprimer de l'hôte
```

1.7 encrypter-décrypter, signer-vérifier

```
tpm2_rsaencrypt -c child -s rsaes clearfile -o encryptedfile
tpm2_rsadecrypt -c child -s rsaes encryptedfile -o clearfile
tpm2_sign -c child -g sha256 -o file.sign file
tpm2_verifysignature -c child -g sha256 -s file.sign -m file
```

1.8 Registres PCR

```
tpm2_pcrreset 0
tpm2_pcrextend 0:sha1=8c83...(hash)
```

1.9 Sauver des données sur le TPM

```
tpm2_evictcontrol -c passwd.ctx 0x81010000 -C o  #sauver
tpm2_unseal -c 0x81010000 > passwd  #récuperer
```

1.10 Sauver des données et protéger avec PCR policy

```
sha1sum passwd #calcul hash
tpm2_pcrreset 0 #flush PCR0
tpm2_pcrextend 0:sha1=8c839... #sauve hash
tpm2_createprimary -C o -G rsa2048 -c primary
```

```
tpm2_startauthsession -S session
tpm2_policypcr -S session -l sha1:0 -L pcr0_policy #créer politique
tpm2_flushcontext session

tpm2_create -C primary -g sha256 \
-u passwd_pcr0.pub -r passwd_pcr0.priv \
-i passwd -L pcr0_policy

tpm2_evictcontrol -c passwd_pcr0 0x81010000 -C o
tpm2_flushcontext session

shred passwd
rm -f passwd

tpm2_startauthsession --policy-session -S session
tpm2_policypcr -S session -l sha1:0
tpm2_unseal -p session:session -c 0x81010000 > passwd
```