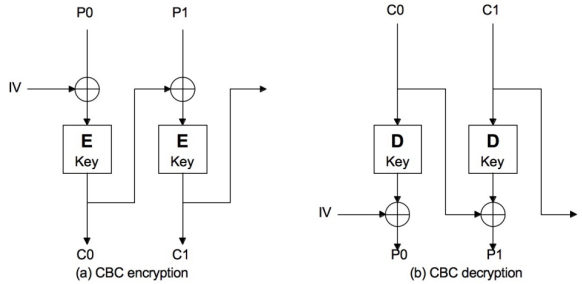
1 TPM

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

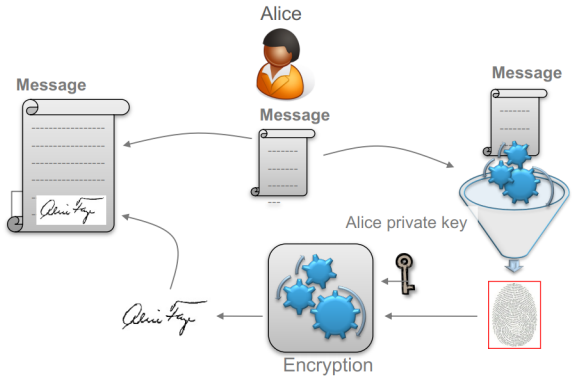
- Deux clés (publique et privée) clé publique disponible par des certificats (CMD pgp)
- Encrypt public -¿ Decrypt private =¿ confidentialité
- Encrypt private-¿ Decrypt public =¿ signature digitale

1.1.3 hash

Transforme un texte, document en un nombre de N bits unique (SHA-2, SHA-3, Blake2).
`md5sum file => a6a0e8d0522e2c5de921b1c455506320` ou `openssl dgst -md5 file MD5(file)= a6a0e8d0522e2c5de921b1c455506320`

1.1.4 signature

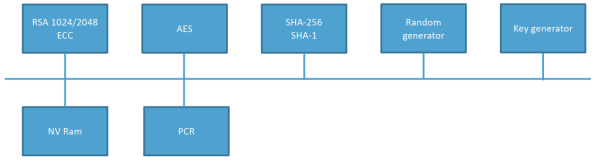
En deux parties: 1. Calcul du HASH puis encryptage avec clé privée.



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

[style=bash]

$tpm2_createprimary - Co - Grsa2048 - co_primcréeruncléprimaireownertpm2_getcaphandles - transientvoirclédanslaRAMtpm2_getcaphandles - persistentvoirclédanslaNV - RAMtpm2_evictcontrol – co_primary.ctxsauveruneclé enNV - RAMtpm2_flushcontext! - teffacertoutelaRAMtpm2_create - Co_prim - Grsa2048 - uchild_pub – rchild_privcréerclé enfant$

$Co_prim - uchild_pub - rchild_priv - cchildchargerclé enfant$

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`