Key derivation

Key derivation

Cipher algorithms require fixed dimension keys

- 56, 128, 256… bits

We may derive keys from multiple sources

- Shared secrets
- Passwords generated by humans
- PIN codes and small length secrets

Original source may have low entropy

- Reduces the difficulty of a brute force attack
- Although we must have some strong relation into a useful key

Sometimes we need multiple keys from the same material

- While not allowing to find the material (a password, another key) from the new key

Key derivation: purposes

Key reinforcement: increase the security of a password

- Usually defined by humans
- Making dictionary attacks impractical

Key expansion: increase the dimension of a key

- Expansion to a size that suits an algorithm
- Eventually derive other related keys for other algorithms (e.g. MAC)

Key derivation

Key derivation requires the existence of:

- A salt which makes the derivation unique
- A difficult problem
- A chosen level of complexity

Computational difficulty

- Transformation requires relevant computational resources

Memory difficulty

- Transformation requires relevant storage resources
- Limits attacks using dedicated hardware accelerators

Key derivation: PKBDF2

Password Based Key Derivation Function 2

Produces a key from a password, with a chosen difficulty

K = PBKDF2(PRF, Salt, rounds, dim, password)

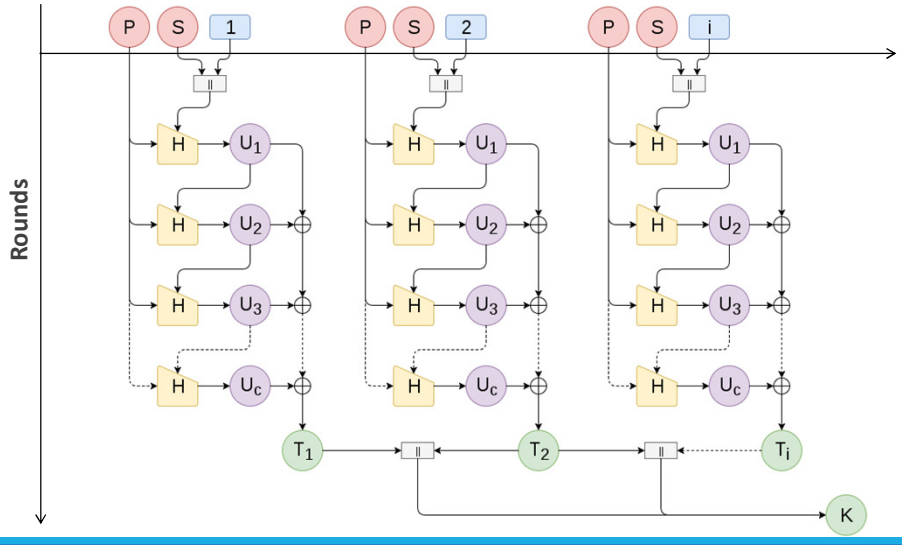
- PRF: Pseudo-Random-Function: a digest function
- Salt: a random value
- Rounds: the computational cost (tens or hundreds of thousands)
- Dim: the size of the result required

Operation: calculates ROUNDS x DIM operations from the PRF using the SALT and PASSWORD

- Larger number of rounds will increase the cost

Key Derivation: PBKDF2

Dimension



Key Derivation: scrypt

Produces a key with a chosen storage cost

K = scrypt(password, salt, n, p, dim, r, hLen, Mflen)

- Password: a secret
- Salt: a random value
- N: the cost parameter
- P: the parallelization parameter. $p \leq (2^{32} - 1) * hLen / MFLen$
- Dim: the size of the result
- R: the size of the blocks to use (default is 8)
- hLen: the size of the digest function (32 for SHA256)
- Mflen: bytes in the internal mix (default is 8 x R)

Key Derivation: scrypt