RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Université d'ORLÉANS

SAE Crypto - Défi 2 : Logarithme discret et attaque Meet-in the-Middle

Ronceray Maxime
REYDET Antonin

Sommaire

Le logarithme Discret - 3

Explication et exemple du fonctionnement du logarithme discret

Diffie-Hellman - 4

Explication du processus Echange de Diffie-Hellman

Meet In The Middle - 5->7

Fonctionnement de l'algorithme meet in the middle

Logarithme Discret

Groupes cycliques $a^{x}(p) = B$

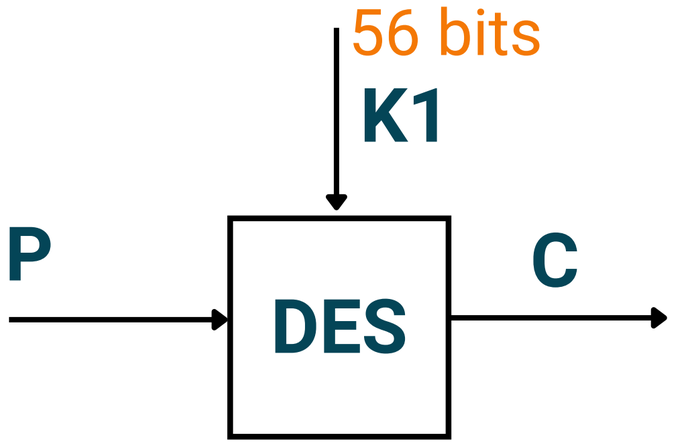
Brute Force

Pour aller plus loin : Le Baby Step / Giant Step

Diffie-Hellman

Alice	Public	BOB
Choisit un entier quelconque	Alice et Bob choisissent : - n un entier quelconque - p un nombre premier	Choisit un entier quelconque
Calcule n [mod p] = Ra		Calcule n^{b} [mod p] = Rb
	Alice et Bob s'échangent Ra & Rb	
Calcule $(R\text{b})$ [mod p] = K_a		Calcule $(R\text{b})^{b}$ [mod p] = K_b

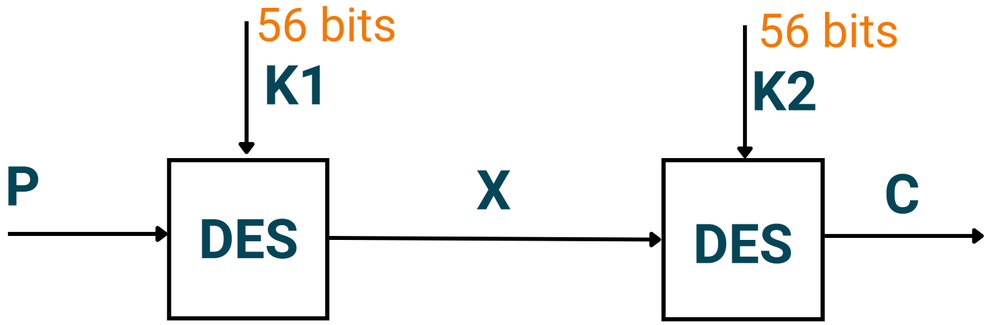
Meet-In-The-Middle (MiTM)



Bruteforce on DES : 2^{56} opérations

Bruteforce on DDES : 2^{112} opérations

MiTM on DDES : 2^{57}



Baby-Step/Giant-Step

Basé sur ré écrire x dans l'équation $a^x(p) = B$

$$a^x(p) = B$$
$$a^{im+j} = B$$
$$a^{im}(a)^j = B$$
$$a^j = B\ (a^{-m})^i$$

m = sqrt(p-1)

0 <= i <m 0 <= j <m

```
p prime : 63691
number x: 9150
solve for x in h = g^x mod p given a prime p
g: 45898
h: 11545
x recovered with bsgs: 9150
```

```python
def bsgs(g, a, p):
    # To solve g^e mod p = a and find e
    m = ceil(sqrt(p-1))
    # Baby Step
    lookup_table = {pow(g, i, p): i for i in range(m)}
    # Giant Step Precomputation c = g^(-m) mod p
    c = pow(g, m*(p-2), p)
    # Giant Step
    for j in range(m):
        x = (a*pow(c, j, p)) % p
        if x in lookup_table:
            return j*m + lookup_table[x]
    return None
```

Temps de résolution d’un problème de logarithme discret selon la méthode d’approche et la longueur de p

+

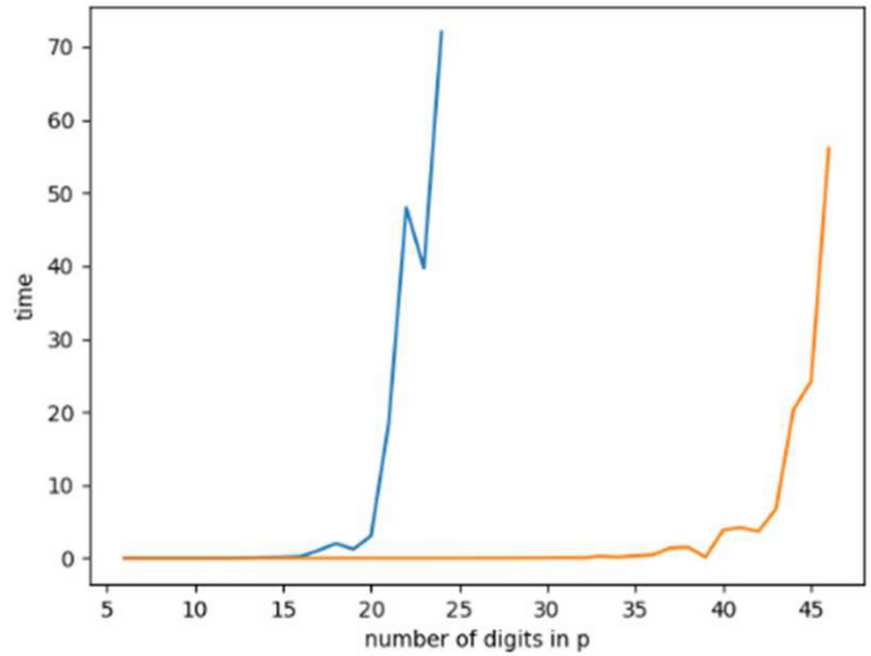
Plus rapide

Multithreading simple

-

Complexe

Stockage



Brute Force **MITM**

CONCLUSION

Au coeur de la cryptographie moderne, ces concepts sont utilisés dans de nombreux protocoles de cryptographie a clé publique. Il est important de continuer a travailler sur ces problèmes afin de les perfectionner et faire progresser la cryptographie

Réferences

[1]. Changyu Dong (S.D). Math in Network Security: A Crash Course [En-ligne]. Disponible : https://www.doc.ic.ac.uk/~mrh/330tutor/index.html

[2]. Ginni(2021). What is Discrete Logarithmic Problem in Information Security? [En-ligne]. Disponible : https://www.tutorialspoint.com/what-is-discrete-logarithmic-problem-in-information-security

[3]. Henri Cohen (S.D) A Course in Computational Algebraic Number Theory.

[4]. Ashutosh Ahelleya (S.D) DLPcand Baby Step Giant Step Algorithm [En-ligne]. Disponible : https://masterpessimistaa.wordpress.com/2018/01/14/dlp-and-baby-step-giant-step-algorithm/

[5]. OpenSSL (2021). Diffie-Hellman [En-ligne]. Disponible : https://wiki.openssl.org/index.php/Diffie_Hellman

[6]. Thomas Pornin (2013). Diffie-Hellman and its TLS/SSL usage [En-ligne]. Disponible : https://security.stackexchange.com/questions/41205/diffie-hellman-and-its-tls-ssl-usage

[7]. Art of the Problem (2012). Public key cryptography - Diffie-Hellman Key Exchange (full version)[En-ligne]. Disponible : https://www.youtube.com/watch?v=YEBfamv-_do&ab_channel=ArtoftheProblem

[8] . Christina Boura , Nicolas David , Rachelle Heim Boissier , and María Naya-Plasencia .(2022). Better Steady than Speedy: Full break of SPEEDY-7-192

[9]. Nicolas David . (2022). Differential Meet-In-The-Middle Cryptanalysis

[10]. Chris Kowalczyk (S.D). Meet In The Middle Attack. [En-ligne]. Disponible : http://www.crypto-it.net/eng/attacks/meet-in-the-middle.html

[X].Nicolas David, doctorant en cryptographique a l’INRIA : nicolas.david@inria.fr