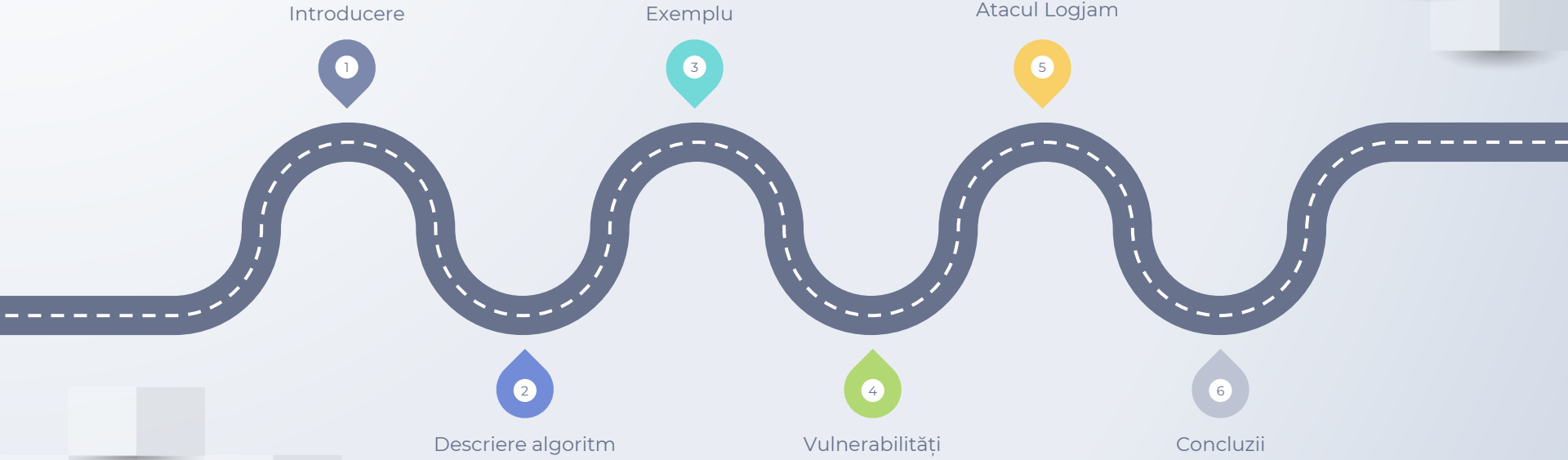
DIFFIE-HELLMAN KEY EXCHANGE

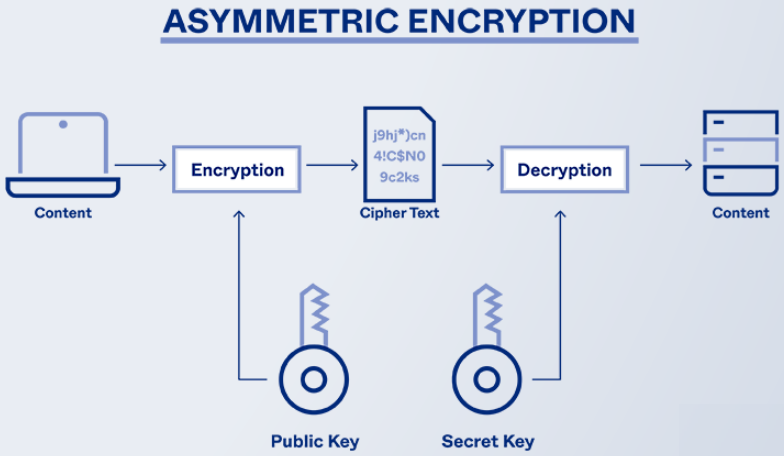
ROADMAP


Introducere
1
Descriere algoritm
2
Exemplu
3
Vulnerabilități
4
Atacul Logjam
5
Concluzii
6

Cum puteți schimba informații în siguranță cu o altă persoană dacă nu ați avut ocazia să împărtășiți cheia înainte?

1. INTRODUCERE

- Diffie-Hellman Key Exchange permite schimbul sigur de chei criptografice între două părți care nu s-au mai întâlnit, printr-un canal public.
- Folosește criptografia asimetrică: există o cheie publică pentru criptare și o cheie privată pentru decriptare. Aceasta permite schimbul de informații într-un mod mai sigur decât în criptografia simetrică, deoarece cheia publică poate fi distribuită liber, iar cheia privată rămâne secretă.
- Utilizat în protocoale de securitate precum TLS (Transport Layer Security), SSH (Secure Shell) și IPsec (IP Security)



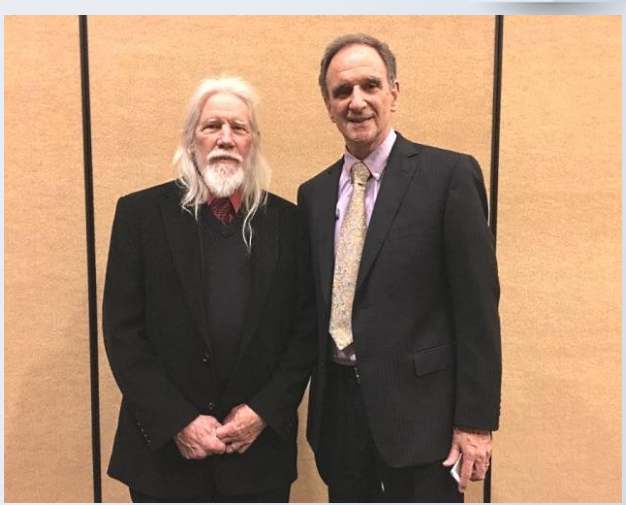
1. INTRODUCERE

Alte întrebuințări:

- Au fost propuse scheme de criptare cu chei publice bazate pe Diffie-Hellman Key Exchange. Prima astfel de schemă este criptarea ElGamal.
- Când doua părți împărtășesc o parolă, acestea pot folosi o formă Diffie-Hellman a password-authenticated key agreement pentru a preveni atacurile de tip man-in-the-middle. Fiecare calculează hash-ul secretului concatenat cu parola, apoi fac schimb și le compară. Dacă se potrivesc, ambele părți au folosit aceeași parolă și pot continua cu acordul cheii. Astfel un atacator poate testa doar o parolă specifică la fiecare iterație cu cealaltă parte, obținând securitate bună cu parole relativ slabe.

1. INTRODUCERE

- Whitfield Diffie, ofițer de securitate la Sun Microsystems și Martin E. Hellman, profesor emerit la Stanford sunt considerați pionierii criptografiei moderne.
- Au câștigat Premiu A.M. Turing în 2015, alături de 1 milion de dolari, pentru contribuțiile semnificative în domeniu.
- Schimbul de chei care le poartă numele a fost descris în lucrarea “New Directions in Cryptography”, publicată în 1976



And the prestigious Nobel Prize in computer science goes to...

2.DESCRIEREA ALGORITMULUI

1. Setup inițial: Alice și Bob aleg împreună două numere întregi pozitive, p și g, unde p este un număr prim, iar g este generatorul lui p (2≤g≤p-2). g trebuie sa fie o rădăcină primitivă modulo p.
Pentru a verifica condiția de rădăcină primitivă, rezultatele:

$g \bmod p$
$g^2 \bmod p$
$g^3 \bmod p$
....
$g^{p-1} \bmod p$

sunt unice și reprezintă o permutare a numerelor cuprinse între 1 si p-1

2. Fiecare alege o cheie privată secretă: a pentru Alice și b pentru Bob, cu 1≤a≤p-2 și 1≤b≤p-2.
Apoi calculează cheile publice folosind formulele: $A = g^a \bmod p$ pentru Alice și $B = g^b \bmod p$ pentru Bob.

3. Ei își partajează cheile publice printr-un mediu nesigur, cum ar fi internetul.

4. Ambii calculează o cheie secretă comună K folosind cheile publice ale celuilalt:
Alice calculează $K = B^a \bmod p$, iar Bob $K = A^b \bmod p$.

Cheile personale a și b nu sunt transmise, făcând aproape imposibilă ghicirea cheii secrete K de către un atacator!

2. EXEMPLU

Alice

1. Setup
p = 17
g = 3

Bob

2.Cheie secretă privată
a = 15

4.Cheie secretă privată
b = 13

3. Cheie publică
$A = 3^{15} \bmod 17 = 6$

5. Cheie publică
$B = 3^{13} \bmod 17 = 12$

6. Combină B cu cheia privată a
$K = 12^{15} \bmod 17 = 10$

7. Combină A cu cheia privată b
$K = 6^{13} \bmod 17 = 10$

8. Cheia secretă comună

Cum rezultă aceeași cheie secretă?
$12^{15} \bmod 17 = 6^{13} \bmod 17$

$(3^{13})^{15} \bmod 17 = (3^{15})^{13} \bmod 17$

2. EXEMPLU CU MAI MULTI PARTICIPANTI

1. Se aleg *p* si *g* de catre Alice, Bob si Carol.
2. Fiecare isi genereaza cheia privata: *a*, *b*, si *c*.
3. Alice calculeaza $g^a \bmod p$ si ii trimite lui Bob.
4. Bob calculeaza $(g^a)^b \bmod p = g^{ab} \bmod p$ si ii trimite lui Carol.
5. Carol calculeaza $(g^{ab})^c \bmod p = g^{abc} \bmod p$ si obtine secretul.
6. Bob calculeaza $g^b \bmod p$ si ii trimite lui Carol.
7. Carol calculeaza $(g^b)^c \bmod p = g^{bc} \bmod p$ si ii trimite lui Alice.
8. Alice calculeaza $(g^{bc})^a \bmod p = g^{bca} \bmod p = g^{abc} \bmod p$ si obtine secretul.
9. Carol calculeaza $g^c \bmod p$ si ii trimite lui Alice.
10. Alice calculeaza $(g^c)^a \bmod p = g^{ca} \bmod p$ si ii trimite lui Bob.
11. Bob calculeaza $(g^{ca})^b \bmod p = g^{cab} \bmod p = g^{abc} \bmod p$ si obtine secretul.

Vulnerabilități

Calculul logaritmului discret

Cheia privată poate fi derivată din cheia publică dacă logaritmul discret poate fi calculat rapid, compromițând astfel securitatea într egului sistem. Provocarea este de a determina a din ecuatia $g^a \bmod p = A$, unde g, p, A sunt cunoscuti. Nu exista nicio metoda efectiva si eficienta cunoscuta, iar pentru p cu minim 600 cifre, devine imposibil

Reutilizarea grupurilor DH și parametrilor comuni

Utilizarea acelorași grupuri de parametri (numărul prim p și generatorul g) de către mai multe sisteme permite atacatorilor să efectueze pre-computări masive și să spargă mai ușor cheile.

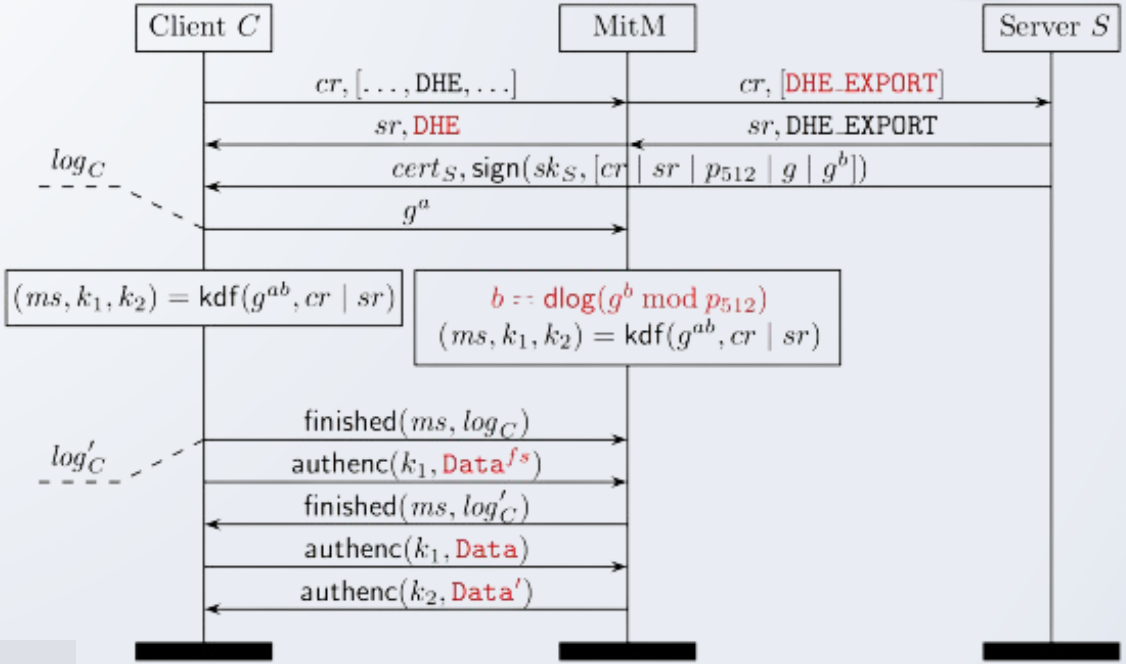
Lungimea insuficientă a cheilor

Utilizarea lungimilor de chei mai scurte, cum ar fi chei de 1024 biți, crește vulnerabilitatea la atacuri criptanalitice avansate, precum calculul eficient al logaritmului discret.

Atacul Logjam

- Atacul Logjam este de tip man-in-the-middle și vizează protocolul TLS, în special implementările Diffie-Hellman Ephemeral (DHE).
- Atacul forțează negocierea cheilor criptografice de lungime redusă (512 biți), exploatând suportul pentru cifrele mai slabe DHE-EXPORT.
- Cifrele slabe, rezultate din restricțiile de export ale criptografiei din anii '90, permit utilizarea cheilor DH mai scurte, facilitând astfel atacurilor să spargă cheia și să descifreze comunicațiile; odată ce cheia este compromisă, atacatorul poate intercepta și altera datele transmise între client și server.
- Pentru a contracara atacul Logjam, este important ca serverele și clienții să dezactiveze suportul pentru cifrele DHE-EXPORT și să folosească seturi de cifre mai puternice.

Prezentare grafică - Atac Logjam



Site-uri web, servere de e-mail și alte servicii dependente de TLS care suportă DHE_EXPORT sunt în pericol de atac Logjam.
Folosind scanări la nivelul întregului internet, s-au obținut următoarele estimări:

Protocol	Vulnerable %
HTTPS – Top 1 Million Domains	8.4%
HTTPS -- Browser Trusted Sites	3.4%
SMTP+StartTLS – IPv4 Address Space	14.8%
POP3S – IPv4 Address Space	8.9%
IMAPS – IPv4 Address Space	8.4%

Atacul Logjam

- Un alt mod de a preveni acest tip de atac este utilizarea criptografiei eliptice, precum Elliptic-curve Diffie–Hellman (nu se cunoaste niciun atac similar pentru aceasta)
- Profitand de structura algebrică a curbelor eliptice, atinge un nivel similar de securitate cu o dimensiune mai mică a cheii. O cheie de 224 de biți oferă același nivel de securitate ca o cheie RSA de 2048 de biți. Acest lucru poate face schimburile mai eficiente și reduce cerințele de stocare.

Concluzii

- Diffie-Hellman a fost văzut ca o inovație, la inventare, în anii 70, ajutând două părți necunoscute să comunice în siguranță. În prezent, noi versiuni mai sigure au apărut, precum Elliptic Curve Diffie-Hellman, folosit la deviceuri IoT și mobile, dar și Ephemeral Diffie-Hellman, folosit de protocolul TLS.
- Este o parte fundamentală a schimbului securizat de date online (la baza TLS). Atât timp cât este implementat împreună cu o metodă de autentificare adecvată, iar p si g au fost selectate corespunzător, nu este considerat vulnerabil la atac.
- Baza pe problema logaritmului discret îl face sigur, dar progresele în Quantum Computing, se așteaptă să rezolve astfel de probleme matematice. Sunt explorați noi algoritmi criptografici rezistenți la atacurile computerelor cuantice, iar aceștia pot înlocui sau spori protocoalele existente în viitor. Spre exemplu, supersingular isogeny key exchange, varianta a Diffie–Hellman, a fost spartă în iulie 2022.

BIBLIOGRAFIE

[1] E. Kehoe, “Diffie and Hellman Receive 2015 Turing Award,” *Notices of the American Mathematical Society*, vol. 63, no. 06, pp. 668–669, Jun. 2016, doi: https://doi.org/10.1090/noti1398.
[2] J. Lake, “Demystifying Diffie-Hellman key exchange and explaining how it works,” *Comparitech.com*, Mar. 15, 2019. https://www.comparitech.com/blog/information-security/diffie-hellman-key-exchange/ (accessed Jan. 02, 2024).
[3] A. S. Gillis, “Diffie-Hellman key exchange (exponential key exchange),” *Security*, 2022. https://www.techtarget.com/searchsecurity/definition/Diffie-Hellman-key-exchange (accessed Jan. 02, 2024).
[4] Baivab Kumar Jena, “Guide to the Diffie-Hellman Key Exchange Algorithm & its Working,” *Simplilearn.com*, Dec. 2021. https://www.simplilearn.com/tutorials/cryptography-tutorial/deffie-hellman-key-exchange (accessed Jan. 02, 2024).

BIBLIOGRAFIE

[5] Revuelto, Vicente, and Krzysztof Socha. "Weaknesses in diffie-hellman key exchange protocol." CERT-EU Computer Emergency Response Team, EE. UU (2016).
[6] Adrian, David, et al. "Imperfect forward secrecy: How Diffie-Hellman fails in practice." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015.
[7] Bokslag, Wouter. "The problem of popular primes: Logjam." *arXiv preprint arXiv:1602.02396* (2016).