UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Schimb de cheie Diffie-Hellman cu certificare IBE

propusă de

Cosmin Ambroci

Sesiunea: *iulie, 2018*

Coordonator științific

Prof.dr. Ferucio Laurențiu Țiplea

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Schimb de cheie Diffie-Hellman cu certificare IBE

Cosmin Ambroci

Sesiunea: *iulie, 2018*

Coordonator științific

Prof.dr. Ferucio Laurențiu Țiplea

Avizat,

Îndrumător Lucrare de Licenţă

Titlul, Numele şi prenumele _____

Data _____ Semnătura _____

DECLARAŢIE privind originalitatea conţinutului lucrării de licenţă

Subsemnatul(a) ……

domiciliul în ……..

născut(ă) la data de ……., identificat prin CNP ……., absolvent(a) al(a) Universităţii „Alexandru Ioan Cuza" din Iaşi, Facultatea de …….. specializarea ……., promoţia ……., declar pe propria răspundere, cunoscând consecinţele falsului în declaraţii în sensul art. 326 din Noul Cod Penal şi dispoziţiile Legii Educaţiei Naţionale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licenţă cu titlul: _____

_____

_____elaborată sub îndrumarea dl. / d-na _____, pe care urmează să o susţină în faţa comisiei este originală, îmi aparţine şi îmi asum conţinutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licenţă să fie verificată prin orice modalitate legală pentru confirmarea originalităţii, consimţind inclusiv la introducerea conţinutului său într-o bază de date în acest scop.

Am luat la cunoştinţă despre faptul că este interzisă comercializarea de lucrări ştiinţifice in vederea facilitării falsificării de către cumpărător a calităţii de autor al unei lucrări de licenţă, de diploma sau de disertaţie şi în acest sens, declar pe proprie răspundere că lucrarea de faţă nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, …… Semnătură student ……

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Schimb de cheie Diffie-Hellman cu certificare IBE*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Prenume Nume*

_____

(semnătura în original)

Cuprins

INTRODUCERE **1**

Tema abordată 1

Soluția propusă 1

Structura lucrări 1

CONTRIBUȚII **2**

1 FORMULAREA PROBLEMEI **2**

1.1 Metoda Diffie-Hellman de schimb de cheie 2

1.1.1 Prezentare generală *2*

1.1.2 Utilizare *3*

1.2 Metoda Diffie-Hellman cu certificare IBE 5

2 DESCRIEREA TEHNICĂ A CONCEPTELOR UTILIZATE **5**

2.1 Descriere tehnica Diffie-Hellman 5

2.2 Descriere IBE 7

2.3 Schema Cocks IBE 8

2.4 Advanced Encryption Standard (AES) 9

3 IBE-DH **12**

3.1 Descrierea aplicației 12

3.2 Securitate 16

3.3 Corectitudine 16

3.4 Complexitate 17

4 COMPARAȚII CU ALTE SISTEME **17**

CONCLUZII **18**

BIBLIOGRAFIE **19**

INDEX DE IMAGINI: **20**

Introducere

Tema abordată

Tema abordată în această lucrare se referă la o nouă metodă de schimb de chei. Protocoalele actuale, precum SSL&TLS, IPSec etc, folosesc diferite variații ale metodei Diffie – Hellman (DH) de schimb de chei. Metoda Diffie – Hellman este utilizată în diverse variante, iar varianta ce oferă autentificare este cea bazată pe certificate. Se cunoaște că utilizarea certificatelor implica folosirea unei infrastructuri cu chei publice (funcții hash, algoritmi de semnare, algoritmi de verificare a semnăturii, formate speciale pentru certificate etc). În general infrastructurile cu chei publice îngreunează foarte mult lucrul cu certificate (datorită verificării semnăturilor digitale care uneori necesită verificarea în lanț a acestora). Cum criptografia bazata pe identitate își propune eliminarea infrastructurii de certificare prin utilizarea a unui simplu generator de chei private (local unui intranet) este natural atunci să ne punem problema construirii unei variante DH cu criptare bazata pe identitate. După cunoștințele noastre o astfel de variantă nu a mai fost propusă in literatura de specialitate. Ca urmare prezenta lucrare propune metoda de schimb de cheie IBE-DH în care utilizarea certificatelor este înlocuită cu criptare bazata pe identitate.

Soluția propusă

Cum protocoalele actuale se bazează pe certificate sau metode derivate din algoritmul Diffie – Hellman (Diffie – Hellman anonim, Diffie – Hellman fix sau Diffie – Hellman efemer), metoda propusă de noi se bazează atât pe Schema IBE a lui Cocks cât și pe algoritmul DH. Prin DH se calculează secretul comun (cheia ce va ajuta în criptările ulterioare) iar prin Cocks IBE se face schimbul necesar pentru DH (schimb ce se va face criptat). Astfel prin această nouă abordare se evită problemele ridicate de actualele protocoale cum ar fi: verificarea lanțurilor de încredere, incapabilitatea de a asigura în același timp atât integritatea, confidențialitatea și autenticitatea mesajelor.

Structura lucrări

Prezenta lucrare este structurată în 4 capitole. Astfel în primul capitol (Formularea problemei) vom discuta despre metoda Diffie – Hellman de schimb de cheie, dar și despre o noua abordare a acestei metode, numita Diffie – Hellman cu certificare IBE (metoda care stă la baza protocolului IBE-DH). Capitolul al doilea are rolul de a explica conceptele folosite în realizarea lucrării, astfel ușurând înțelegerea acesteia. În al treilea capitol am descris pe larg metoda propusă atât din punct

de vedere al aplicării cât și al corectitudinii, securității și complexități oferite. În ultimul capitol se face o comparație dintre IBE-DH și protocoale DH de schimb de cheie deja existente

Contribuții

Așa cum s-a menționat deja lucrarea noastră propune o noua metoda de schimb de cheie bazata pe protocolul DH, și anume metoda DH cu criptare bazată pe identitate (IBE-DH). Pentru criptarea bazată pe identitate am folosit schema Cocks, cunoscută ca fiind eficientă si ușor de implementat. Alte metode de schimb de cheie se bazează ori pe latici de întregi, ori pe aplicații biliniare. Ambele variante necesită efort semnificativ de prezentare. Ca urmare implementarea IBE-DH prin schema Cocks este fără îndoială cea mai eficientă. Concluzionăm acest paragraf prin a menționa încă o dată că după cunoștințele noastre, aceasta este prima încercare din literatura de specialitate de a utiliza metoda DH cu criptare bazata pe identitate.

1 Formularea problemei

În acest capitol vom face o descriere completă a metodei DH și vom formula precis problema noastră: definirea metodei DH cu criptare bazată pe identitate prin schema cocks.

1.1 Metoda Diffie-Hellman de schimb de cheie

1.1.1 Prezentare generală

Metoda Diffie-Hellman de schimb de cheie (prescurtată DH) este o metodă criptografică securizată de schimb de chei într-un canal public de comunicare. A fost unul din primele protocoale de schimb de chei din domeniul criptografic. Metoda DH permite ca doi utilizatori ce nu au informații unul despre celălalt să ajungă la o aceiași cheie secretă comună. Această cheie va fi folosită pentru criptarea mesajelor într-un sistem de criptare cu chei simetrice. Schema a fost publicată pentru prima dată de Whitfield Diffie și Martin Hellman în 1976 în articolul „New directions in cryptography”. Deși se pare că acest protocol a fost descoperit câțiva ani mai devreme de către James Ellis, Clifford Cocks și Malcolm Williamson, dar a fost ținut secret.

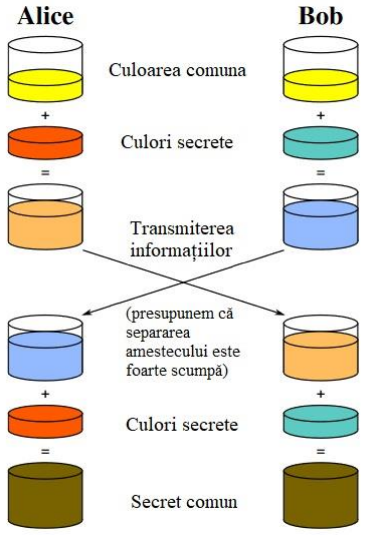


Fig. 1 - Schema DH folosind culori

Protocolul stabilește un secret comun ce va fi folosit pentru comunicarea într-o rețea publică. Fig. 1 are rolul de a explica ideea generală din spatele algoritmului folosind culori în locul numerelor foarte mari. Punctul crucial este ca Alice și Bob să schimbe culorile secrete doar amestecate. La final se obține exact aceeași culoare (cheie), care este dificil din punct de vedere computațional (imposibil pentru calculatoarele moderne sa realizeze calculele într-un timp rezonabil) de obținut de către o persoana ce ascultă canalul. (1)

1.1.2 Utilizare

Algoritmul DH a fost folosit în multe protocoale de securitate. Cele mai importante dintre acestea sunt: *Security Sockets Layer (SSL), Secure Shell (SSH), IP Security (IPSec)*

I. Security Sockets Layer (SSL)

SSL este un standard de securitate dezvoltat de Netscape în 1994 pentru a stabili o conexiune criptată între un server și un browser. Aceasta conexiune asigura confidențialitate și integritate pentru toate datele ce trec de la browser la server și invers.

SSL folosește certificate, perechi de chei publice și private, chei DH pentru a oferi confidențialitate (*schimb de chei*), autenticitatea și integritatea este asigurata prin *Message Authentication Code (MAC)*. Aceste informații fac parte din *Public Key Infrastructure (PKI)*. SSL este utilizat în special pentru traficul de date business și financiar. Este posibil ca un utilizator să nu observe că se folosește SSL în timpul comunicării, dar este foarte posibil sa observe unele întârzierii.

SSL/TLS este alcătuit din două straturi: stratul inferior, numit *protocolul de înregistrare*, folosește TCP pentru a administra criptografia simetrică (privată) și stratul superior, numit *protocolul de handshake,* se folosește de DH. Protocolul handshake permite server-ului să se autentifice clientului folosind un protocol cu chei publice. De asemenea facilitează cooperarea server-ului cu clientul pentru crearea unei chei simetrice ce va fi folosită pentru a cripta și decripta rapid.

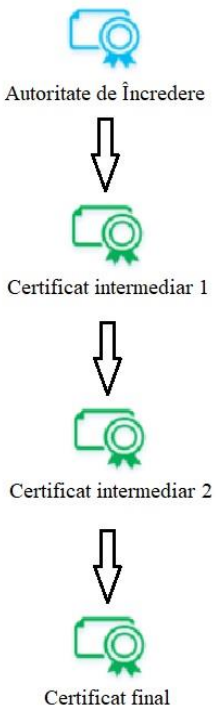
II. Secure Shell (SSH)

SSH este un protocol de rețea foarte des întâlnit ce asigura securitate autentificărilor pe internet. Acest protocol a apărut ca înlocuitor pentru Telnet, ce este fără securitate, și FTP, deoarece și Telnet și FTP nu criptează datele, ele fiind trimise în clar. SSH-ul criptează, autentifică și comprimă datele transmise automat.

Protocolul SSH se realizează în trei faze. În prima fază, numită „Hello”, server-ul transmite o listă de algoritmi suportați. Această lista detaliază perechile de chei DH. În a doua fază cele două parți ajung la o aceiași cheie secretă *x* prin intermediul protocolului DH. În ultima fază se folosește cheia *x* pentru a cripta o alta cheie (cheia aplicației) ce va fi folosită pentru criptarea tuturor mesajelor ce urmează a fi transmise.

III. IP Security (IPSec)

IPSec este o extensie a Internet Protocol (IP) – ce este o suită de protocoale introdusă de Internet Engineering Task Force (IETF) pentru a ajuta la configurarea unui canal de comunicații între mașini multiple. La fel ca protocoalele anterioare, IPSec folosește DH și criptografia asimetrică pentru a stabili identități, algoritmii folosiți și un secret comun. Înainte ca IPSec să înceapă criptarea sunt necesare câteva schimburi de informații. Componenta IPSec care facilitează aceste schimburi se numește Internet Key Exchange (IKE). IKE folosește DH pentru a stabili un secret comun folosind mecanismele clasice, secret ce va fi folosit mai departe în procesul de criptare. Cheia nu va fi schimbata printr-un canal nesigur. (2)



Majoritatea protocoalelor care folosesc DH au nevoie de certificate și de lanțuri de încredere (Fig. 2). Un lanț de încredere este o listă în care se găsesc certificate intermediare și certificate ale autorităților. Lanțul se termina cu un certificat al rădăcinii, acest certificat este mereu semnat de către el însuși. Înainte ca un certificat să fie de încredere trebuie să fie semnat de o autoritate de încredere. În momentul în care se face verificarea certificatului și se observă că nu e semnat de o autoritate credibilă, se merge mai departe în lanț și se verifică urmatoarele certificate. Se repetă această operație până se găsește un certificat semnat de o autoritate de încredere, caz în care se poate stabili o conexiune sigură. Dacă nu se găsește un certificat emis de o autoritate de încredere în tot lanțul, conexiunea este abandonată. (3)

Fig. 2 - Exemplu de lanț de încredere

1.2 Metoda Diffie-Hellman cu certificare IBE

Propunem o nouă abordare, DH fără utilizarea certificatelor standard (X.509), dar cu certificare IBE. Criptografia cu chei publice se bazează pe o pereche de chei (o cheie publică și cheie privată) pentru a cripta și decripta mesaje. Cheia privată este deținută doar de către posesor. În majoritatea cazurilor cheia publică se găsesc în interiorul unor certificate și este disponibilă tuturor utilizatorilor autorizați. Standardul X.509 descrie cum informația este trecuta în interiorul unui certificat. Certificatele sunt emise de către o autoritate. În momentul în care ele sunt utilizate se verifica semnătura entității ce le-a emis. Gradul de încredere într-un certificat este gradul de încredere în entitatea ce la emis. Datorită acestor certificate se pot forma lanțuri de încredere (Fig. 2 - Exemplu de lanț de încredere). (4)

Metoda propusă evită utilizarea certificatelor clasice, certificarea folosită fiind asigurata de către schema IBE. Astfel acest tip de protocol nu necesită un certificat semnat de o autoritate superioară, certificarea fiind făcută în baza identității. Acest lucru este posibil datorită faptului că IBE se folosește tocmai de identitate ca de o cheie publică atunci când vrea să trimită un mesaj. Identitatea fiind alcătuită din adresa de e-mail (furnizată la înscrierea în sistem) concatenată cu o dată[1] (un exemplu de cheie poate fi: exemplu@exemplu.exemplu-28.08.2018-02:00). Pentru că valabilitatea certificatului în protocolul IBE este variabilă, acesta poate fi schimbat oricând fără a afecta corectitudinea. În metoda propusă de noi, certificatul ceste schimbat la fiecare înregistrare.

2 Descrierea tehnică a conceptelor utilizate

În acest capitol o să vorbim despre principalele concepte care intră în alcătuirea protocolului, astfel o înțelegere a acestuia fii mai ușoară. Aici o să fie prezentate următoarele concepte: algoritmul DH, problema logaritmului discret, schemă IBE, Cocks IBE și AES.

2.1 Descriere tehnica Diffie-Hellman

După cum am văzut în Fig. 1 cu ajutorul acestui algoritm se poate ajunge la o aceeași cheie comună. Cea mai simpla metoda de implementare, care de altfel este si metoda inițială, folosește un grup de numere întregi modulo p, unde p este un număr prim. În continuare avem un exemplu de aplicare a protocolului.

- Alice și Bob primesc numerele prime p = 23 și g = 5

[1] Poate conține data când mesajul poate fi citit, ora (astfel limitând timpul de valabilitate) sau o combinație a acestora.

- Alice alege un număr întreg ca și secret a = 6, după îi trimite lui Bob A = $g^a \ mod \ p$
  - A = $5^6 \ mod \ 23 = 8$
- Bob alege un număr întreg ca și secret b =15, după îi trimite lui Alice B = $g^b \ mod \ p$
  - A = $5^{15} \ mod \ 23 = 19$
- Alice calculează s = $B^a \ mod \ p$
  - s = $19^6 \ mod \ 23 = 2$
- Bob calculează s = $A^b \ mod \ p$
  - s = $8^{15} \ mod \ 23 = 2$

Bob și Alice au ajuns la un aceiași valoare deoarece $(g^a)^b = (g^b)^a$. Se poate observa că doar a și b sunt ținuți secreți, celelalte valori fiind publice. O dată ce Alice și Bob au ajuns la un același secret îl pot folosi pe post de cheie de criptare pentru a transmite mesaje într-un canal de comunicare deschis. Bine înțeles *a, b* și *p* trebuie să aibă valori mult mai mari pentru a putea face acest exemplu sigur (exista doar 23 de răspunsuri posibile pentru ecuația *n* mod 23). Dacă valoarea primă *p* ar avea cel puțin 300 de cifre și *a*, respectiv *b*, ar avea cel puțin 100 de cifre, atunci chiar și cel mai puternic/rapid calculator nu ar putea (într-un timp rezonabil) să găsească *a* cunoscând *g, p,* $g^b \ mod \ p$ și $g^a mod \ p$. Problema pe care calculatorul ar trebui să o rezolve se numește *problema logaritmului discret*. Calculul $g^a mod \ p$ este cunoscut ca *exponențiere modulară* și poate fi făcut în mod eficient chiar și pentru numerele foarte mari. Se poate observa că *g* nu e nevoie să fie foarte mare, în practică el este un număr prim mic (2, 3, 5, 7…) deoarece el este folosit ca bază în exponențiere. Cum urmează să fie ridicat la o putere mare, va rezulta o valoare foarte mare de aici rezultând un număr mare de rădăcini primitive.

Această schemă poate fi folosită pentru un număr oarecare de mare. Spre exemplu Alice, Bob și Carol pot aplica acest algoritm după cum urmează:

1) Părțile stabilesc parametri p și g
2) Părțile își stabilesc secretele
3) Alice calculează g^a și îl trimite lui Bob
4) Bob calculează $(g^a)^b = g^{ab}$ și îl trimite către Carol
5) Carol calculează $(g^{ab})^c = g^{abc}$ și îl păstrează
6) Bob calculează g^b și îl trimite lui Carol
7) Carol calculează $(g^b)^c = g^{bc}$ și îl trimite către Alice
8) Alice calculează $(g^{bc})^a = g^{abc}$ și îl păstrează

9) Carol calculează g^c și îl trimite lui Alice
10) Alice calculează $(g^c)^a = g^{ca}$ și îl trimite către Bob
11) Bob calculează $(g^{ca})^b = g^{abc}$ și îl păstrează

Metoda generală care a fost prezentată mai sus constă în ridicarea lui g la toate secretele utilizatorilor. Cea mai ușoară metoda de aplicare și de altfel cea mai evidentă este de a pune toți cei N utilizatori în cerc și a face câte o rotație, începând pe rând cu fiecare. Această metodă presupune ca fiecare utilizator să facă N exponențieri modulare rezultând o complexitate de N^2. Alegând o ordonare și o parcurgere mai bună și bazându-ne pe faptul ca unele chei pot fi duplicate, este posibil sa reducem numărul de exponențieri modulare a fiecărui participant la $\log_2(N) + 1$ folosind metoda divide et impera.

Securitatea acestui algoritm se bazează pe problema logaritmului discret. Astfel fie p un număr prim, iar a o rădăcină primitivă de ordinul $p - 1$ a unității. Aceste două valori fiind fixate, problema logaritmului se poate reformula astfel:

Fiind dat un $\beta \in Z_p^*$, să se determine exponentul $a \in Z_{p-1}$ astfel ca $\alpha^a \equiv \beta \pmod p$.

Evident această problemă se poate rezolva printr-o căutare directă (forță brută – calculăm toate puterile lui α). Această căutare de tip exhaustiv se realizează în timp $O(p)$ și folosind $O(1)$ memorie. Pe de-altă part, dacă se calculează anterior într-o tabelă toate valorile $(a, \alpha^a \bmod p)$, aflarea valorii căutate se poate face în $O(1)$, dar cu un spațiu de complexitate $O(p)$. Toți algoritmi construiți pentru calculul logaritmului discret stabilesc un compromis spațiu – timp (1).

2.2 Descriere IBE

O schemă de Criptare Bazată pe Identitate (en. Identity Base Encryption – prescurtată IBE) este un criptosistem în care orice șir de caractere poate fi o cheie publică validă. În practică se folosește o adresă de e-mail și o dată. Sistemele IBE care folosesc e-mail-ul prezintă urmatoarele avantaje:

- Expeditorul poate trimite un mesaj unui destinatar ce încă nu și-a stabilit cheia publică
- Înainte să trimită un mesaj expeditorul nu trebuie să facă o verificare activă pentru a obține parola destinatarului
- Expeditorul poate trimite un mesaj ce va fi citit la un moment dat în viitor
- Sistemul actualizează automat cheia privată la o anumită perioadă de timp

Din 1984 de când a fost căutată o schemă de criptare cu chei publice în care cheia publică poate fi un șir de caractere arbitrar au fost câteva propuneri pentru aceasta schemă. Totuși nici una nu era pe deplin satisfăcătoare. Unele cereau ca utilizatorii să nu colaboreze, altele aveau nevoie de mult prea mult timp pentru a genera cheia privată. Este drept să spunem că un criptosistem IBE utilizabil a fost o problemă deschisă. Performanțele protocolul IBE bazat pe e-mail sunt asemănătoare cu cele produse de criptarea ElGamal. Securitatea sistemului este oferita de o analogie cu calculul pe curbe eliptice. (5)

2.3 Schema Cocks IBE

Schema Cocks IBE este un sistem de criptare bazat pe identitate propus de Clifford Cocks în 2001. Securitatea acestui sistem este asigurată de problema reziduurilor pătratice. Protocolul se realizează în patru pași:

- Setarea:

  Se calculează o valoare $n = pq$, unde p și q sunt doua numere prime, și o valoare e astfel încât $e \in J_n - QR_n$ și se fac publici parametrii: n, e și h, unde h este o funcție hash

- Extragerea:

  Se calculează $a = h(ID)$, unde ID reprezintă identitatea utilizatorului. Dacă $a \in QR(n)$, cheia privată (r) se va alege dintre rădăcinile pătrate ale lui a, altfel cheia privată (r) se va alege dintre rădăcinile pătrate ale lui ea

- Criptare:

  Fie $a = h(ID)$. Pentru criptarea unui bit $m \in \{-1,1\}$, alegem aleatoriu $t_1, t_2 \in Z_n^*$ astfel încât $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = m$, unde $\left(\frac{t_1}{n}\right)$ reprezintă simbolul Jacobi[2] dintre t_1 și n. După se calculează $c_1 = t_1 + at_1^{-1} mod\ n$ și $c_2 = t_2 + eat_2^{-1} mod\ n$. Din c_1 și c_2 se formează o pereche ce reprezintă criptotextul asociat bitului m.

- Decriptare:

  Dacă $r^2 \equiv a\ mod\ n$ atunci $c = c_1$, altfel $c = c_2$. Bitul m se decriptează astfel: $m = \left(\frac{c+2r}{n}\right)$.

[2] Pe parcursul lucrării, notația pentru simbol Jacobi va fi $\left(\frac{t}{n}\right)$. Simbolul Jacobi este echivalent cu Simbolul Legendre, dacă n este prim

După cum se poate vedea algoritmul IBE Cocks criptează bit cu bit și fiecare bit este criptat prin $2 \log n$ biți. Astfel se poate vedea că această schemă nu este cea mai bună alegere pentru a cripta mesaje. Dar după cum a observat și Cocks schema poate fi folosită în practică pentru transmiterea cheilor de sesiune, fapt ce transformă schema in una folositoare. (6)

2.4 Advanced Encryption Standard (AES)

În septembrie 1997, National Institute of Standard and Technology (NIST) a pornit o competiție în urma căreia algoritmul câștigător urma să devină următorul standard de criptare. Competiția a fost demarată deoarece s-a observat că actualul standard DES (Data Encryption Standard) a început să fie vulnerabil la atacurile de tip forță brută, acest lucru fiind posibil datorită avansului tehnologic și faptului că DES folosea chei pe 64 de biți. NIST a declarat că ei caută un criptosistem bloc la fel de sigur ca și 3DES (se aplică DES de trei ori peste mesaj), dar mult mai eficient. Cerința minimă de funcționare pentru criptosistemele bloc o reprezenta lucrul cu blocuri de 128 de biți și chei cu lungimea de 128, 196 și 256 biți. În urma cerințelor ridicate cerute de NIST mulți candidați au renunțat, astfel la sfârșitul primei etape rămânând 5 potențiali candidați: MARS, RC6, Rijndael, Serpent și Twofish. (7)

În ianuarie 1999 distributet.net și Electronic Frontier Foundation au colaborat public pentru a demonstra că se poate sparge o cheie DES în mai puțin de 24 de ore prin forță brută. Experimentul acesta a fost un succes, cheia fiind spartă în 22 de ore și 15 minute, aducând vulnerabilitățile algoritmului la lumină.

În octombrie 2000 NIST a anunțat câștigătorul competiției ca fiind Rijndael, algoritm dezvoltat de V. Rijmen și J. Daemen. Acest algoritm a fost ulterior denumit Advanced Encryption Standard (AES). Începând cu anul 2002 algoritmul a fost adoptat ca standard de criptare pentru guvernul SUA și a fost introdus în ISO/IEC 18033-3[3], iar în iunie 2003 este adoptat ca și standard de criptare pentru informațiile clasificate. AES este mult mai sigur decât predecesorii lui (DES și 3DES) datorită folosirii unei chei de o lungime mai mare (128, 192, 256 biți). Acest algoritm poate fi găsit în majoritatea protocoalelor de securitate a rețelelor (ex. SSL), în majoritatea aplicațiilor și dispozitivelor moderne ce au nevoie de securitate.

Există trei tipuri de AES: AES-128, AES-192 și AES-256. Fiecare algoritm criptează și decriptează blocuri de date de 128 de biți folosind chei de 128, 196 și respectiv 256 biți. Algoritmul inițial

[3] Standard de criptare ce prevede criptosistemele bloc

(Rijndael) a fost proiectat pentru a folosi si alte dimensiuni ale mesajului și cheii, dar în implementarea finala a AES-ului s-a renunțat la aceste obținui. Criptosistemul folosește aceeași cheie atât pentru criptare cât și pentru decriptare, asta implicând ca cealaltă persoană să aibă aceeași cheie (8).

Algoritmul AES definește o serie de transformări (permutări și substituții) care sunt efectuate asupra datelor, date stucate în memorie sub formă de vector. Spre deosebire de DES, AES realizează transformările într-un număr variabil de runde, număr ce depinde de lungimea cheii. Pentru chei de 128 de biți sunt 10 runde, pentru chei de 196 de biți sunt 12 runde, iar pentru chei de 256 de biți sunt necesare 14 runde. La fiecare rundă se folosește o altă cheie derivată din cea inițială. Când vorbim despre criptosisteme bloc, trebuie să ne gândim și la metodele de criptare. Aceste metode sunt: Electronic Codebook (ECB), Chiper Block Chaining (CBC), Propagating CBC (PCBC), Chiper Feedback (CFB), Output Feedback (OFB) și Counter (CTR).

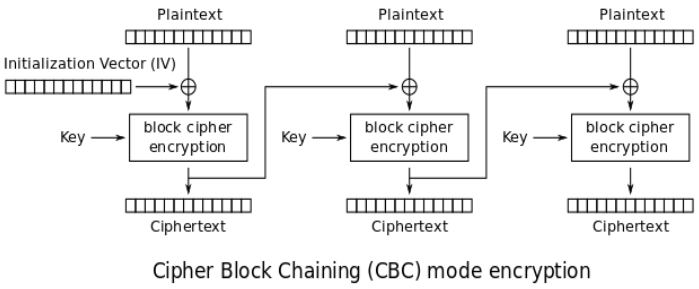


Fig. 3 - Schemă pentru o criptare de tip CBC

În continuare o să vorbim puțin despre aceste metode.

- ECB – Este cea mai simplă metoda. Ea constă în împărțirea mesajului în blocuri și criptarea fiecărui bloc în parte. Marele dezavantaj îl constituie faptul că dacă există doua blocuri la fel, ele o să fie criptate identic. Acest lucru produce o scurgere de informații.

- CBC – Este o metodă asemănătoare cu ECB, doar că aici se aplică o operație XOR între blocul ce urmează să fie criptat și blocul criptat anterior (după cum se poate vedea și în Fig. 3). Este una dintre cele mai folosite metode.
- PCBC – Această metodă lucrează ca și metoda CBC doar că aici există două operații XOR. Una între blocul de text anterior și criptotextul rezultat și cealaltă operație va fi între rezultatul primului XOR și blocul actual ce urmează să fie criptat.
- CFB – Este o metoda derivată din CBC, deoarece se face o criptare și o operație XOR, dar în următoarea ordine: inițial luăm blocul criptat anterior, îl criptam cu cheia specifică, iar asupra rezultatului aplicăm operația XOR cu blocul corespunzător obținut din mesaj.

- OFB – Această tehnică este aproape identică cu tehnica CFB deoarece ordinea operațiilor este aceeași, doar că aici blocul care va fi criptat nu mai este rezultatul operațiilor anterioare, ci este blocul obținut imediat după criptare. Pentru lămuriri am introdus următoarea schemă:

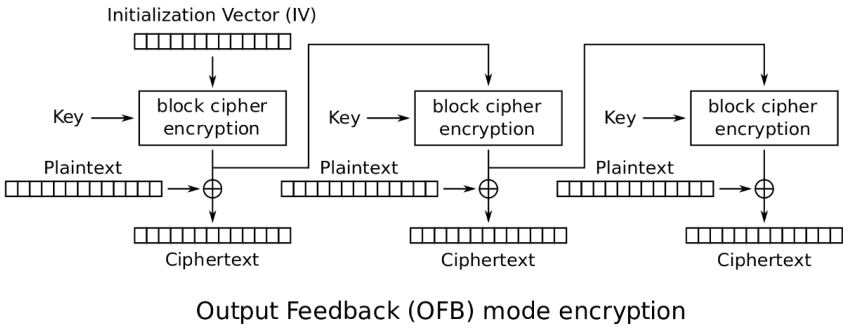
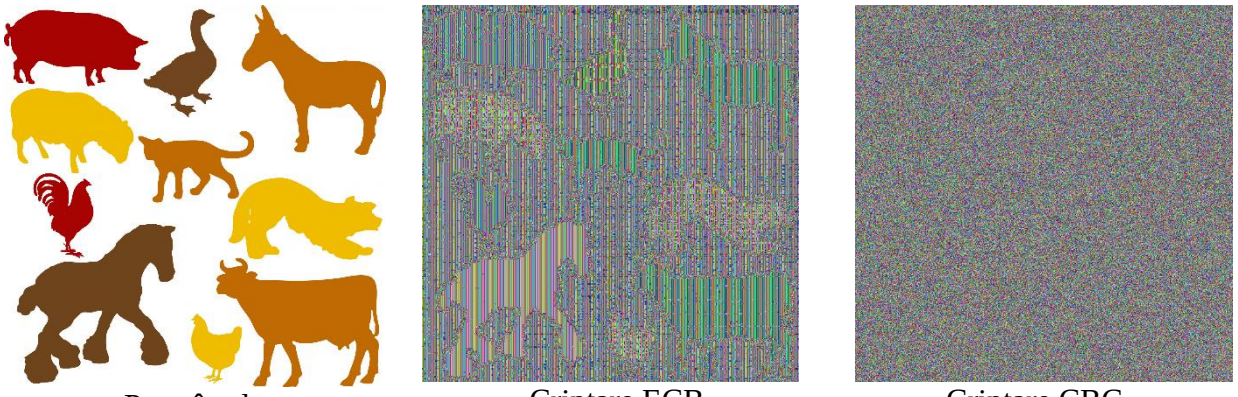


Fig. 4 - Metoda de criptare OFB

- CTR – Această tehnică este diferită de celelalte abordări. Ea se aplică astfel: se generează un număr aleatoriu (nonce) și se concatenează cu numărul blocului ce urmează sa fie criptat, valoarea obținută din această concatenare se criptează și apoi se aplică operația XOR între blocul proaspăt criptat și blocul din textul în clar.

După cum se poate vedea în Fig. 5 scurgerile pa care le produce metoda de criptare ECB sunt eliminate prin celelalte metode (pentru exemplificare a fost folosit CBC), scurgeri ce pot dezvălui foarte mult din mesaj.



Poza în clar | Criptare ECB | Criptare CBC

Fig. 5 - Diferențe între criptări

3 IBE-DH

În acest capitol vom prezenta detaliat caracteristicile protocolului propus. Acesta a fost implementat într-o aplicație de chat. Datorită avantajelor pe care protocolul le oferă, schimbul de mesaje este sigur.

3.1 Descrierea aplicației

Aplicația propriu-zisă reprezintă un sistem de chat de nivel de bază în care am urmărit îmbinarea mai multor sisteme de securitate mai vechi pentru a asigura confidențialitatea, integritatea și autenticitatea mesajelor. Aplicația a fost realizată în Python 3.6 pe modelul client – server. Este alcătuită dintr-un server general și o serie de utilizatori ce o să comunice între ei. Serverul are rolul de a stabili parametri de securitate necesari protocoalelor de criptare, de a rezolva unele cereri ale utilizatorilor, de a ști ce utilizatori sunt activi și cum pot fi aceștia contactați. Aplicația utilizatorului are rolul de a facilita comunicarea în interiorul aplicației. Ideea în jurul căreia s-a format toată această aplicație este de a lua Schema lui Cocks și de a o îmbunătăți pentru a putea fi utilizata în practică. Din ce știm din paragraful 2.3 Schema lui Cocks poate fi folosita pentru criptarea mesajelor de mici dimensiuni deoarece aceasta criptează bit cu bit, iar fiecare bit este criptat în doua elemente. Modul în care aplicația funcționează este prezentat schematic în Fig. 6.

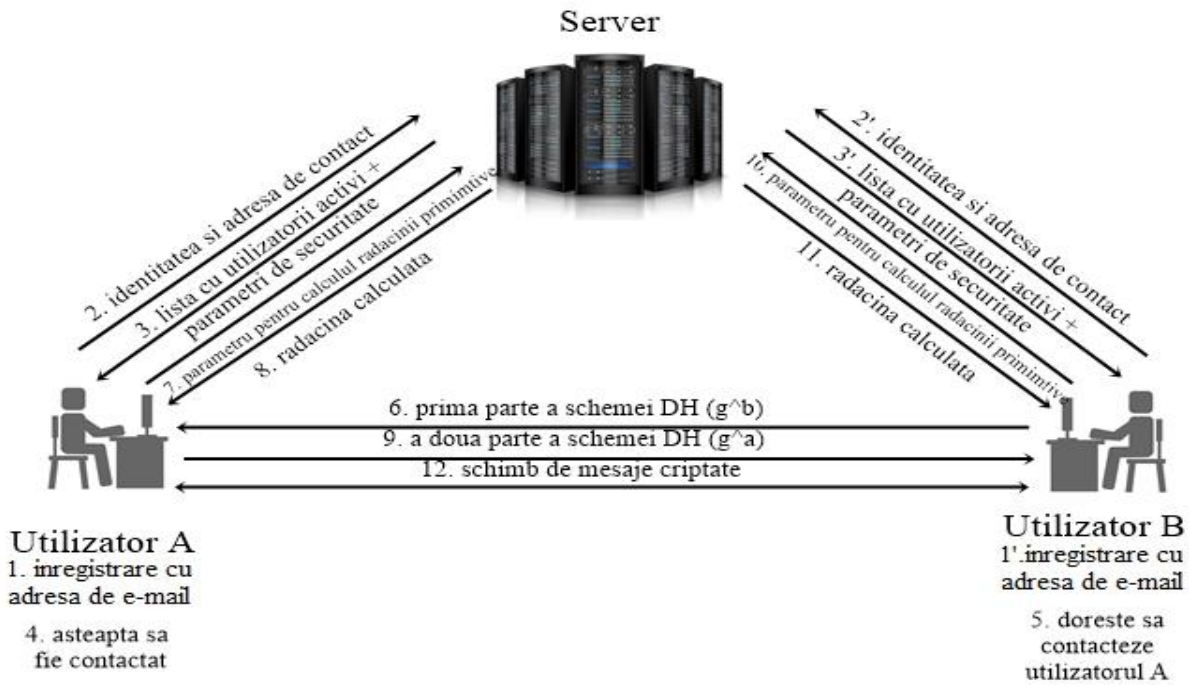


Fig. 6 - Schema generală a protocolului

În primul rând o să considerăm că serverul principal este pornit, astfel parametrii de securitate generali ai aplicației sunt stabiliți. Acești parametri sunt alcătuiți din trei numere prime generate aleatoriu, astfel: p și q pe 128 de biți și $g \in (0, 256)$ și un reziduu pătratic modulo n ($n = p * q$) numit e. În acest moment serverul poate accepta conexiuni, datorită faptului că este un server de tip concurent el poate accepta mai multe conexiuni în același timp. În schemă acest lucru este reprezentat prin folosirea ’. Când utilizatorul deschide aplicația trebuie să își introducă identitatea (identitatea este adresa de e-mail) și o parolă. Parola va avea un rol important în timpul folosirii algoritmului DH, identitatea fiind necesară pentru identificarea interlocutorului, dar și în timpul Schemei Cocks.

Protocolou IBE-DH se realizează între două entități ce vor să comunice. Pentru o înțelegere mai ușoară o să facem identificrea entităților prin emițător și receptor. Acesta are următoarea structură:

1) Emițătorul anunță receptorul că urmează să fie contactat de cineva. Anunțul se face prin trimiterea identității emițătorului.

   Pentru o implementare riguroasă receptorului trebuie să interogheze serverul cu privire la identitatea primită pentru a se asigura că acesta știe de ea sporind gradul de încredere în această conexiune. În cazul în care serverul nu ar confirma identitatea, încercarea de conectare este respinsă.

2) Protocolul transformă parola emițătorului într-un număr ($password\_number\_e$) astfel:
   I. Se creează un șir în care fiecare componentă este reprezentarea Unicode a caracterului respectiv;
   II. Se adună toate valorile într-o variabilă.

   Pentru implementare pot exista diferite variații ale acestui calcul, el având rolul de a transforma un șir de caractere într-un număr.

3) În continuare se face o interogare la server pentru a transforma identitatea receptorului mesajelor într-un număr. Această transformare are loc astfel:
   I. Se aplică o funcție hash de tip sha256 peste identitate;
   II. Dacă numărul întreg obținut ($hash\_identity$) are cel puțin o rădăcină pătratică $mod\ n$, atunci se returnează $hash\_identity$;
   III. Dacă $hash\_identity$ nu are nici o rădăcină pătratică $mod\ n$, atunci se mai aplică încă o dată funcția hash dar peste $hash\_identity$;

IV. Se repetă pașii II și III până când se poate returna o valoare.

4) Emițătorul trimite numărul proaspăt obținut de la server, către receptor.
5) Se calculează valoarea $number\_to\_encript = g^{password\_number\_e}$. Pentru o implementare mai riguroasă calculul poate avea mici variații. Un exemplu de calcul ar putea fi acesta: $number\_to\_encript = g^{password\_number\_e} mod\ m$, unde m este un număr pe maxim 16 biți.
6) În continuare se calculează o serie de tuple[4] astfel:
   I. $number\_to\_encript$ este transformat în biți și puși într-o listă ($binary\_number\_password$);
   II. Se parcurge $bynary\_number\_password$ și se transformă toate valorile de 0 în -1;
   III. Pentru fiecare element (m) din $binary\_number\_password$ se generează aleatoriu două numere t_1 și t_2 astfel încât $\left(\frac{t_1}{n}\right) = \left(\frac{t_2}{n}\right) = m$, unde $\left(\frac{t_1}{n}\right)$ reprezintă simbolul Jacobi dintre t_1 și n;
   IV. Dacă t_1 și t_2 respectă condiția, programul calculează $reverse\_t_1 = t_1^{-1}$ și $reverse\_t_2 = t_2^{-1}$, unde t_1^{-1} este inversul $mod\ n$ al lui t_1;
   V. Dacă nu respectă se reaplică pasul III până aceasta este îndeplinită;
   VI. După ce au fost calculate inversele modulare urmează calculul efectiv al componentelor tuplei corespunzătoare bitului astfel:
      a. Prima componentă (c_1) se află așa: $c_1 = t_1 + hash\_identity * reverse\_t_1\ mod\ n$;
      b. A doua componentă (c_2) se calculează astfel: $c_2 = t_2 + e * hash\_identity * reverse\_t_2^{-1}\ mod\ n$.

   După cum se poate observa aici se folosește criptarea din Schema lui Cocks (2.3).

7) După calcularea tuturor tuplelor acestea o să fie transmise una câte una către receptor. Acum emițătorului va aștepta primirea următoarelor informații: identitatea cu care receptorul criptează tuplele ce urmează să le trimită și tuplele pentru a fi decriptate.
8) În acest moment receptorul a primit toate tuplele și urmează decriptarea lor astfel:
   I. Se face o interogare la server pentru ca acesta să îi calculeze o rădăcină primitivă $hash\_identity\ mod\ n$. Acest calcul o să fie făcut astfel:

[4] Un tuplu este un grup de două înregistrări.

a. Serverul calculează $jacobi\_p = \left(\frac{hash\_identity}{p}\right)$;

b. Dacă $jacobi\_p$ este egal cu 1 atunci se calculează reziduul pătratic ($square\_root$) dintre $hash\_identity$ și n și se returnează această valoare împreună cu eticheta cu valoarea 1;

c. Dacă $jacobi\_p$ este egal cu -1 atunci se calculează reziduul pătratic dintre $e * hash\_identity$ și n și se returnează această valoare împreună cu eticheta cu valoarea 2;

II. În funcție de eticheta primită receptorul urmând să decripteze primul element din fiecare tuplu sau al doilea element astfel:

a. Dacă eticheta este 1 atunci valoarea bitului corespunzător tuplei este $\left(\frac{c_1 + 2 * square\_root}{n}\right)$

b. Dacă eticheta este 2 atunci valoarea bitului corespunzător tuplei este $\left(\frac{c_2 + 2 * square\_root}{n}\right)$

c. Valorile obținute se adaugă într-o listă care urmează să fie alterată prin schimbarea tuturor valorilor de -1 în 0

d. După înlocuire se transformă șirul de biți obținut într-un număr.

După cum se poate vedea această metodă de transmitere este cea descrisă de către Schema Cocks IBE (2.3).

9) Acum receptorul deține valoarea $g^{password\_number\_e}$.
10) În continuare se vor repeta pașii de la 2) la 9), dar cu mențiunea că rolurile se vor inversa o singură dată: receptorul este acum emițătorul și invers.
11) În momentul în care emițătorul deține $g^{password\_number\_r}$ și receptorul deține $g^{password\_number\_e}$ aplicația emițătorului va calcula: $cheie\_e = (g^{password\_number\_r})^{password\_number\_e}$ și aplicația receptorului va calcula: $cheie\_r = (g^{password\_number\_e})^{password\_number\_r}$. După cum se poate vedea cei doi au ajuns la aceiași cheie ($cheie\_e = cheie\_r$). Astfel am folosit metoda DH (1.1) pentru a ajunge la o cheie comună fără a face publice sau a transmite secretele fiecărui utilizator.
12) Când emițătorul a ajuns la $cheie\_e$ și receptorul la $cheie\_r$ se începe schimbul efectiv de mesaje criptate folosind algoritmul AES datorită faptului că părțile au o cheie comună.

Pentru a exemplifica funcționarea protocolului IBE-DH, am realizat o implementare a sa. Aceasta constă în realizarea unei aplicații de chat care utilizează protocolul propus de noi pentru a asigura integritatea, securitatea și confidențialitatea mesajelor transmise între doi utilizatori, conform punctelor descrise mai sus.

3.2 Securitate

Securitatea algoritmului este asigurată de către securitatea metodei DH și a schemei Cocks IBE.

- Securitatea metodei DH este asigurată de dificultatea problemei logaritmului discret și de faptul că valoarea exponentului DH este informație privata a uniu utilizator. Valoarea DH partajată de doi utilizatori este de forma $g^{ab} mod\ p$ și aceasta valoare nu poate fi determinată prin algoritmi polinomiali determiniști din informațiile: g, p, g^b mod p și g^a mod p. Aceasta datorită faptului că determinarea unuia dintre exponenți a sau b este o instanță a lemei logaritmului discret iar determinarea lui $g^{ab} mod\ p$ din $g^a mod\ p$ și $g^b mod\ p$ este o instanță a problemei DH (cunoscută ca fiind de asemenea grea).
- Securitatea schemei Cocks se bazează pe dificultatea stabilirii daca un element $x \in J_n$ este reziduu pătratic sau nu. Aceasta problemă este considerată grea la momentul actual. Ca urmare este dificil a decide între cele două componente ale unui criptotext cocks care este componenta corectă, altfel spus cu probabilitate de $\frac{1}{2}$ oricare din cele două componente poate fi cea corectă. Dacă mesajul transmis are n biți, atunci probabilitatea ca o anumită secvență să fie mesajul corect transmis este $\frac{1}{2^n}$. Pentru n suficient de mare aceasta este o probabilitate neglijabilă, la aceasta se mai adaugă faptul că cunoașterea componentei corecte necesită și determinarea unei rădăcini pătrate modulo un număr compus. Aceasta problemă este de asemenea grea, considerând că factorizarea nu poate fi făcută in timp polinomial.

3.3 Corectitudine

Corectitudinea algoritmului este asigurată de către corectitudinea metodei DH și schemei Cocks IBE.

- După cum am arătat în capitolul 2.1 metoda DH are la bază două exponențieri ale aceluiași număr (g). În orice ordine am face aceste două exponențieri observăm, că ajungem la

același rezultat ($(g^a)^b = (g^b)^a$). Atâta timp cât nu apar alterări externe, calculele vor fi corecte.

- Corectitudinea algoritmului Cocks IBE reiese din:

$$\left(\frac{c_1 + 2 * r}{n}\right) = \left(\frac{t_1 + a * t_1^{-1} + 2 * r}{n}\right) = \left(\frac{t_1^{-1} * (t_1^2 + r^2 + 2 * r * t_1)}{n}\right) = \left(\frac{t_1^{-1} * (t_1 + r)^2}{n}\right) =$$

$$= \left(\frac{t_1^{-1}}{n}\right) * \underbrace{\left(\frac{(t_1 + r)^2}{n}\right)}_{=1} = \left(\frac{t_1^{-1}}{n}\right) = \left(\frac{t_1}{n}\right) = m$$

3.4 Complexitate

Complexitatea schemei IBE-DH este obținută din complexitatea metodei DH, care este $O\ (log^3 n)$, la care se adaugă complexitatea schemei IBE, care este $O\ (log^2\ n)$ în cazul nostru (prin utilizarea schemei Cocks). Ca urmare, complexitatea schemei propuse nu este mai mare decât complexitatea schemei DH cu certificate, care, pe lângă complexitatea schemei DH implică complexitatea verificării semnăturilor digitale ce este $O\ (log^3\ n)$. Însă, IBE-DH elimina in totalitate verificarea certificatelor și, eventual, utilizarea unui lanț de încredere pentru semnatarul certificatelor (care ar putea implica verificarea unui lanț de certificate).

4 Comparații cu alte sisteme

Protocolul IBE-DH se poate compara cu variantele DH utilizate în IPSec și SSL&TLS.

În IPSec se folosește unul din următoarele trei metode de criptare: AES, DES sau 3DES[5]. Mesajele sunt semnate prin una din metodele: MD5, SHA-1, SHA-2. Pentru schimbul de cheie se folosește algoritmul DH. După stabilirea cheii IPSec poate trimite mesajele în două moduri: AH (Authentication Header) sau ESP (Encapsulating Security Payload). Metoda AH are rolul de a oferi autenticitate mesajelor, fără criptarea efectivă a lor. Metoda ESP are rolul de criptare și autentificare a mesajelor. La primirea mesajelor se face o verificare pentru a se vedea dacă nu cumva pachetele au fost alterate. În majoritatea implementărilor algoritmului se folosește metoda ESP deoarece este mai sigură. (9)

În SSL (ca și detaliere a punctului I din 1.1.2) parametri de securitate a schemei (metoda de schimb de chei, metoda de criptare, algoritmul MAC) sunt stabiliți după primul mesaj „Hello” al serverului,

[5] Este tot algoritmul DES în care se folosesc 3 chei. Este folosit pentru plusul de siguranță pe care în aduce. Criptotextul se obține prin criptarea mesajului cu prima cheie, decriptarea cu a doua cheie și recriptare cu a treia cheie.

astfel existând două tipuri de calcule criptografice: criptografie asimetrică (folosita pentru a stabili o cheie comună ce va fi folosita pentru criptare) și criptografie simetrică (folosită pentru criptarea efectivă a mesajelor). Pentru calculele criptografiei asimetrice este utilizat unul din următoarele algoritme: RSA, DH sau FORTEZZA. Dezavantajul pe care îl prezintă aceste metode în cardul protocolului SSL este faptul că cheile publice trebuie certificate de către o autoritate, lucru ce conduce la formarea de lanțuri de încredere. După cum se poate observa, și acest protocol se lovește tot de problema certificatelor, a lanțurilor de încredere și a semnării datelor în partea de stabile a secretului comun.

Comparativ cu aceste metode algoritmul IBE-DH face mai puține operați (nu face semnări și verificări ale semnăturilor, care pot conduce la verificarea unui întreg lanț de încredere) datorită faptului că autenticitatea mesajelor este asigurată de schema IBE folosită. Pe lângă faptul că complexitatea metodei IBE-DH este ușor mai bună decât a metode DH cu certificate, eliminarea verificării certificatelor aduce un plus major de eficienta practica.

Concluzii

În această lucrare am propus o metodă nouă de schimb de cheie, IBE-DH. Ea se alătură variantelor DH utilizate in protocoale de genul IPSec sau SSL&TLS, adăugând o noutate majora: eliminarea certificatelor. Ca urmare, metoda noastră utilizează criptografie bazată pe identitate, în care cheia publică se construiește din identitatea destinatarului. Materialul de cheie utilizat de cele 2 părți ce comunica se construiește pe baza unor informații distribuite de un generator de cheie. Comparativ cu versiunea DH cu certificate, metoda noastră este ușor mai eficienta în sensul că, are complexitatea $log^2 n + log^3 n$ în loc de $2 * log^3 n$. Mai mult decât atât, se elimina verificarea oricarui lanț de încredere asociat certificatelor. Ca o posibilă extensie a metodei, se pot utiliza alte tipuri de scheme bazate pe identitate, precum schemele bazate pe aplicații biliniare sau latici. Metoda IBE-DH constituie la momentul actual subiectul publicației (10)

Bibliografie

1. **Tarau, Paul.** Diffie–Hellman key exchange. s.l., Texas : University of North Texas, 28 October 2014.

2. **Ahmed, Maryam, și alții.** *Diffie-Hellman and Its Application in Security Protocols.*, International Journal of Engineering Science and Innovative Technology, p. 72, 2012.

3. What is the SSL Certificate Chain? *dnsimple.* [Interactiv] https://support.dnsimple.com/articles/what-is-ssl-certificate-chain/.

4. X.509 Public Key Certificates. *Microsoft.* [Interactiv] https://msdn.microsoft.com/en-us/library/windows/desktop/bb540819(v=vs.85).aspx.

5. IBE Secure E-mail. *IBE Secure E-mail.* [Interactiv] 2002. https://crypto.stanford.edu/ibe/#description.

6. **Țiplea, Ferucio Laurențiu, și alții.** The Cocks PKE and IBE Schemes. *Security of Identity-based Encryption Schemes.* 2016. Vol. LNCS 10006.

7. **Daemen, Joan și Rijmen, Vincent.** *The Design of Rijndael.* s.l. : Springer, 2002.

8. **Rouse, Margaret.** Advanced Encryption Standard (AES) . *TechTarget.* [Interactiv] Martie 2017. https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard.

9. About IPSec Algorithms and Protocols. *Fireware Help - WatchGuard.* [Interactiv] https://www.watchguard.com/help/docs/fireware/12/en-US/Content/en-US/mvpn/general/ipsec_algorithms_protocols_c.html.

10. **Țiplea, Ferucio Laurențiu, Ambroci, Cosmin și Chistol, Ana-Maria.** Identity Based Encryption Diffie - Hellman Key Exchange. In preperation.

Index de imagini:

Fig. 1 - Schema DH folosind culori ---------- *2*
Fig. 2 - Exemplu de lanț de încredere ---------- *4*
Fig. 3 - Schemă pentru o criptare de tip CBC ---------- *10*
Fig. 4 - Metoda de criptare OFB ---------- *11*
Fig. 5 - Diferențe între criptări ---------- *11*
Fig. 6 - Schema generală a protocolului ---------- *12*

Surse imagini

Fig. 1 https://bit.ly/2Ko8qgR , cu mențiunea că am modificat imaginea

Fig. 2 Creată de mine

Fig. 3 https://bit.ly/2KbLPrP

Fig. 4 https://bit.ly/2Kn0oHW

Fig. 5 https://bit.ly/2tDp1WN, cu mențiunea că doar prima imagine a fost preluată de pe net, celelalte două fiind create de mine prin criptări

Fig. 6 Creată de mine