

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Protocoale de stabilire a cheilor de grup
bazate pe scheme de partajare a secretelor
(Secret Sharing-based
Group Key Establishment)

LUCRARE DE DOCTORAT
REZUMAT

Coordonator:

Prof.univ.dr.Adrian ATANASIU

Doctorand:

Ruxandra-Florentina OLIMID

București
2013

Informații de contact ale autorului:

ruxandra.olimid@fmi.unibuc.ro

Comisia de examinare:

Conf. Dr. Radu Gramatovici (Universitatea din București, Romania - președinte)

Prof. Dr. Adrian Atanasiu (Universitatea din București, Romania - conducător științific)

Prof. Dr. Victor Valeriu Patriciu (Academia Tehnică Militară, Romania - referent național)

Prof. Dr. Ferucio Laurențiu Țiplea (Universitatea din Iași, Romania - referent național)

Prof. Dr. Bogdan Warinschi (Universitatea din Bristol, Anglia - referent internațional)

Abstract

Aplicațiile de grup permit mai multor utilizatori să partajeze resurse sau să desfășoare activități în colaborare, oferind posibilitatea unor drepturi și responsabilități diferențiate în cadrul grupului. Securitatea reprezintă un aspect important pentru astfel de aplicații, fiind dificil de implementat, mai ales când dimensiunea grupului este mare și membrii acestuia sunt localizați în zone geografice sau de rețea diferite și beneficiază de mecanisme de protecție diverse. Pentru a dobândi proprietăți criptografice de bază (precum confidențialitate, autenticitate și integritate) de obicei este necesar ca membrii grupului să stabilească o cheie secretă comună. Aceasta se obține în urma execuției unui *protocol de stabilire a cheilor de grup*.

Teza se rezumă la studiul protocoalelor de stabilire a cheilor de grup bazate pe *scheme de partajare a secretelor*, un mecanism criptografic care permite divizarea unui secret în mai multe componente astfel încât numai submulțimi autorizate de astfel de componente să permită reconstrucția. Deși partajarea secretelor aduce multiple avantaje când este utilizată în cadrul protocoalelor de stabilire a cheilor de grup, există în momentul de față două deficiențe principale: (1) multe protocoale nesigure au fost publicate în ultimii ani și (2) foarte puține construcții se bazează pe demonstrații riguroase de securitate.

Prima parte a tezei se concentrează pe scheme de partajare a secretelor. Rezumă o abordare non-clasică a acestora, definește o nouă schemă vizuală de partajare și analizează posibilitatea construcției malițioase a sistemelor de partajare. A doua parte se axează pe protocoale de stabilire a cheilor de grup care utilizează scheme de partajare a secretelor. Introduce multiple atacuri asupra protocoalelor recent publicate, subliniind astfel necesitatea demonstrațiilor de securitate. Rezumă proprietățile necesare pentru atingerea unui nivel de securitate optim și analizează pe scurt metode de analiză a securității. În final, introduce un protocol de punere de acord asupra cheii comune de grup care atinge un nivel ridicat de securitate în timp ce menține constant numărul de runde de comunicație (indiferent de dimensiunea grupului).

Tuturor celor care mi-au influențat viața și munca.

Cuprins

Lista de simboluri și notații	7
Prezentare generală și organizare	8
Partea I. Partajarea secretelor	11
1 Introducere în partajarea secretelor	13
2 Scheme de partajare a secretelor	15
2.1 Scheme de m -partajare	15
2.2 O nouă schemă vizuală de partajare	16
3 SETUP asupra SSS care utilizează generatoare de numere aleatoare	21
Partea II. Protocoale de stabilire a cheilor de grup bazate pe scheme de partajare a secretelor	25
4 Introducere în protocoale de stabilire a cheilor de grup	27
5 Protocoale GKE informal sigure bazate pe SSS	29
5.1 Harn și Lin (2010)	29
5.2 Hsu et al. (2012)	29
5.3 Sun et al. (2012)	33
5.4 Yuan et al. (2013)	33
6 Protocoale GKE formal sigure bazate pe SSS	43
6.1 Un nou protocol	43
6.2 Demonstrații de securitate	43
6.3 Analiză	45
Concluzii și direcții viitoare de cercetare	47
Bibliografie	49

Lista de simboluri și notații

$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	=	mulțimea numerelor naturale, întregi, respectiv reale
p, q	=	numere naturale prime
\mathbb{Z}_p	=	mulțimea întregilor modulo $p \in \mathbb{N}$, i.e. $\{0, \dots, p-1\}$
\mathbb{Z}_p^*	=	grupul multiplicativ $\{1, \dots, p-1\}$
m	=	numărul posibil de utilizatori (participanți / membri ai grupului / entități / jucători / părți)
$\mathcal{U} = \{U_1, \dots, U_m\}$	=	mulțimea utilizatorilor posibili
(s_i)	=	sesiunea i (a unui protocol)
t	=	numărul participanților la o sesiune dată
$\mathcal{U}_{(s_i)}$	=	mulțimea participanților la sesiunea (s_i)
$k, k_{(s_i)}$	=	cheia de grup, cheia de grup a sesiunii (s_i)
\mathcal{D}	=	dealerul
\mathcal{AS}	=	structura de acces a unei SSS
S	=	secretul partajat
S^i	=	al i -lea secret partajat
$Sp(S)$	=	mulțimea tuturor secretelor posibile
s_i, s_i^j	=	componenta i a unui secret S , componenta i a unui secret S^j
$Sp(s)$	=	mulțimea tuturor componentelor posibile
$S(x, y)$	=	pixelul (x, y) al imaginii secrete S (în cazul VSSS)
$s_i(x, y)$	=	pixelul (x, y) al componentei s_i (în cazul VSSS)
$x \leftarrow^R X$	=	o alegere uniform aleatoare a lui x din mulțimea X
$A \rightarrow B : M$	=	un mesaj M transmis de A lui B
$A \rightarrow^* : M$	=	un mesaj M de tip broadcast transmis de A
H, H_i, h, h_i	=	funcții hash, $i \in \mathbb{N}$
(pk, sk)	=	o pereche cheie publică - cheie privată
(pk_i, sk_i)	=	o pereche cheie publică - cheie privată a utilizatorului U_i
$\Sigma.\text{Sign}_{U_i}(M)$	=	semnătura utilizatorului U_i asupra mesajului M
$\Sigma.\text{Verify}_{U_i}(M, \sigma)$	=	verificarea semnăturii σ a utilizatorului U_i corespunzătoare mesajului M
\mathcal{K}	=	parametrul de securitate
q_r	=	numărul maxim de interogări a unei funcții hash
q_s	=	numărul maxim de sesiuni (concurrente)
$Pr[X]$	=	probabilitatea de realizare a evenimentului X
$\text{Adv}_A^{\{AKE, MA, Con\}}$	=	avantajul adversarului PPT (Probabilistic Polynomial Time) \mathcal{A} împotriva securității AKE , securității MA , respectiv contributivității unui protocol

Prezentare generală și organizare

Partea I se concentrează asupra schemelor de partajare a secretelor, dar se rezumă strict la noțiunile și schemele utilizate pe parcursul lucrării și la contribuțiile personale. Conține următoarele capitole:

- **Capitolul 1** introduce schemele de partajare și clasificarea acestora. Scopul principal constă în familiarizarea cititorului cu noțiunile necesare pentru parcurgerea tezei, fără să aibă ca obiectiv prezentarea unui studiu complet.
- **Capitolul 2** descrie mai întâi două scheme clasice de partajare a secretelor, care ulterior vor fi utilizate în construcția protocoalelor de stabilire a cheilor de grup. Apoi, prezintă perspectiva non-clasică a *schemelor de m-partajare*, o abordare care aduce avantaje suplimentare în cadrul stabilirii cheilor de grup. În final, capitolul introduce o schemă vizuală de partajare a secretelor, analizează proprietățile acestora și prezintă rezultatele obținute în urma implementării.
- **Capitolul 3** extinde SETUP (Secretly Embedded Trapdoor with Universal Protection) la schemele de partajare, arătând vulnerabilitatea acestora în cazul în care utilizează o cantitate suficientă de informație aleatoare. Analizează atacul propus și consideră anumite tehnici de protecție. Cu scopul de a exemplifica utilitatea practică, mecanismul este particularizat pentru schema de partajare Shamir, cea mai populară în literatura de specialitate.

Partea II este centrată asupra protocoalelor de stabilire a cheilor de grup bazate pe partajarea secretelor. În funcție de modul de analiză a securității, protocoalele se împart în: protocoale care beneficiază de o demonstrație formală (riguroasă) de securitate și protocoale care se bazează numai pe argumente informale și incomplete. Conține următoarele capitole:

- **Capitolul 4** introduce protocoalele de stabilire a cheilor de grup și clasificarea acestora în protocoale de transfer și protocoale de punere de acord (agreare). Crează legătura cu schemele de partajare, utilizate în construcția acestor protocoale.
- **Capitolul 5** se concentrează pe protocoalele care nu dețin o demonstrație riguroasă de securitate. Tratează patru astfel de protocoale introduse în ultimii ani (2010-2013), descrie mai multe atacuri asupra acestora și indică posibile metode de îmbunătățire.
- **Capitolul 6** se concentrează asupra protocoalelor care conțin o demonstrație riguroasă de securitate. Mai întâi, descrie noțiunile de bază utilizate în demonstrațiile formale,

tehnica *succesiunii de jocuri* (*sequence of games*), modelul GBG și limitările modelelor de securitate actuale. Apoi, prezintă două protocoale existente care beneficiază de o demonstrație formală de securitate. Contribuția principală a capitolului constă în introducerea unui nou protocol demonstrat sigur în modelul GBG, care menține același număr de runde ca protocoalele existente, dar beneficiază de multiple avantaje datorate schemelor de m -partajare.

În varianta completă a tezei, secțiunea **Contribuții personale** a fiecărui capitol sumează rezultatele proprii și indică lucrările originale (excepție fac Capitolele 1 și 4, care sunt introductive). Rezumatul ignoră această secțiune pentru că se axează pe contribuțiile personale, reducând la minim prezentarea noțiunilor deja existente.

În finalul lucrării, secțiunea **Concluzii și direcții viitoare de cercetare** reia rezultatele personale prezentate și sugerează posibile direcții viitoare de studiu.

Lista de Simboluri și Notății indică notațiile utilizate pe parcursul lucrării. Deși câteodată amintim cititorului să facă referire la această listă, de obicei nu repetăm semnificația simbolurilor în cadrul lucrării. Ca o remarcă generală, omitem în general să specificăm spațiul matematic în care se lucrează, acesta fiind evident din descriere (de exemplu nu menționăm mod p de fiecare dată când operațiile au loc într-un grup finit de ordin p).

Varianta completă a tezei conține două **Anexe**, care rezumă cunoștințele necesare de criptografie și matematică:

- **Anexa A. Noțiuni de bază în Criptografie** descrie primitivele criptografice utilizate pe parcursul lucrării. Considerăm că cititorul este deja familiarizat cu acestea, însă le reamintim pentru completitudine și specificarea notațiilor.
- **Anexa B. Noțiuni de bază în Matematică** prezintă câteva concepte și leme care se aplică în cadrul tezei.

Partea I

Partajarea secretelor

Capitolul 1

Introducere în partajarea secretelor

Definiția 1.1. (*Scheme de partajare a secretelor (SSS¹)*). O schemă de partajare a secretelor este un proces sau un protocol care împarte un secret primit la intrare în componente astfel încât doar mulțimi autorizate de componente permit reconstrucția secretului inițial [8].

Mai formal, este o pereche de algoritmi (Share, Rec), unde:

1. Share(S, m) este un algoritm nedeterminist de partajare care primește la intrare un secret S și scoate la ieșire o mulțime de m componente $\{s_1, \dots, s_m\}$;
2. Rec(s_{i_1}, \dots, s_{i_t}), $t \leq m$ este un algoritm determinist de reconstrucție din componente care scoate la ieșire S dacă $\{s_{i_1}, \dots, s_{i_t}\}$ reprezintă o mulțime autorizată.

Mulțimea tuturor mulțimilor autorizate se numește *structură de acces* și se notează cu \mathcal{AS} . În mod normal, componentele sunt distribuite mai multor participanți, astfel încât ne vom referi la o mulțime *autorizată* de participanți ca fiind o mulțime capabilă să determine secretul prin colaborare. O structură de acces se numește *monotonă* dacă orice mulțime care conține o mulțime autorizată este de asemenea autorizată.

În general, SSS constau în mai multe faze:

1. *Inițializare*. Definește mediul de desfășurare al schemei: parametrii, spațiul secretelor și al componentelor, alte premise.
2. *Partajarea*. Descrie algoritmul de divizare al secretului în componente. În multe cazuri, o entitate de încredere numită *dealer* realizează partajarea.
3. *Distribuția*. Indică componentele transmise fiecărui participant prin canale sigure. În general, fiecare participant primește o singură componentă. Există însă cazuri când o entitate primește mai multe componente (și deci capătă o putere mai mare în cadrul grupului) sau chiar o întreagă mulțime autorizată (și deci devine capabilă să determine secretul fără colaborarea cu alți membri ai grupului²).
4. *Reconstrucția*. Explicitează formulele sau algoritmi necesari pentru determinarea secretului dintr-o mulțime autorizată de componente.

¹Secret Sharing Schemes.

²Acesta este cazul schemelor de m -partajare.

Definiția 1.2. (*Scheme de partajare perfecte*) [8]. O schemă de partajare este *perfectă* dacă componentele corespunzătoare mulțimilor neautorizate nu oferă nici o informație despre secret.

Definiția 1.3. (*Scheme de partajare ideale*) [8]. O schemă de partajare este *ideală* dacă este perfectă și dimensiunea componentelor este egală cu dimensiunea secretului.

Menționăm două tipuri de SSS cu structuri de acces speciale:

- *SSS cu structura de acces de tip prag.* Structura de acces conține toate mulțimile cu cardinalul cel puțin t , $1 \leq t \leq m$. Informal, orice t sau mai multe componente sunt suficiente pentru reconstrucția secretului, în timp ce mai puțin de t componente nu. Numim o astfel de schemă (t, m) -schemă de partajare.
- *SSS Unanime.* Structura de acces conține ca singur element mulțimea tuturor componentelor. Informal, toate componentele sunt necesare pentru reconstrucția secretului.

În funcție de secretul partajat, SSS se diferențiază în:

- *Scheme text.* Secretul partajat este o secvență de biți (ex.: un element al unui corp finit sau un text). Sunt cele mai populare scheme de partajare și prezintă o utilitate crescută în practică.
- *Scheme vizuale.* Secretul partajat este o imagine.
- *Scheme audio sau video.* Secretul partajat este un fișier audio sau video.

Deși există multiple alte clasificări ale SSS (ex.: dinamice vs. statice, care partajează un singur secret vs. care partajează secrete multiple, proactive vs. reactive, cu dealer vs. fără dealer, etc.) și proprietăți speciale (ex.: detecția și identificarea trișorilor, verificabilitate) ne rezumăm strict la noțiunile necesare pentru lucrarea de față.

Capitolul 2

Scheme de partajare a secretelor

2.1 Scheme de m -partajare

Definiția 2.1. (Scheme de m -partajare). O schemă de m -partajare este o schemă care împarte un secret de m ori în același număr de componente t utilizând aceeași schemă de partajare de bază (i.e. aceiași algoritmi Share și Rec).

Altfel spus, o schemă de m -partajare execută de m ori o schemă de bază pentru același secret S și același număr de componente t . Fie \mathcal{AS} structura de acces a schemei de m -partajare și \mathcal{AS}_i , $i = 1, \dots, m$ structurile de acces corespunzătoare execuțiilor schemei de bază. Se observă imediat că:

$$\mathcal{AS}_1 \cup \dots \cup \mathcal{AS}_n \subseteq \mathcal{AS}. \quad (2.1)$$

Definiția 2.2. (Scheme de m -partajare perfecte) [20]. O schemă de m -partajare este perfectă dacă se satisfac următoarele condiții: (1) structura sa de acces este dată de $\mathcal{AS} = \mathcal{AS}_1 \cup \dots \cup \mathcal{AS}_n$; (2) mulțimile neautorizate nu oferă nici o informație despre secret.

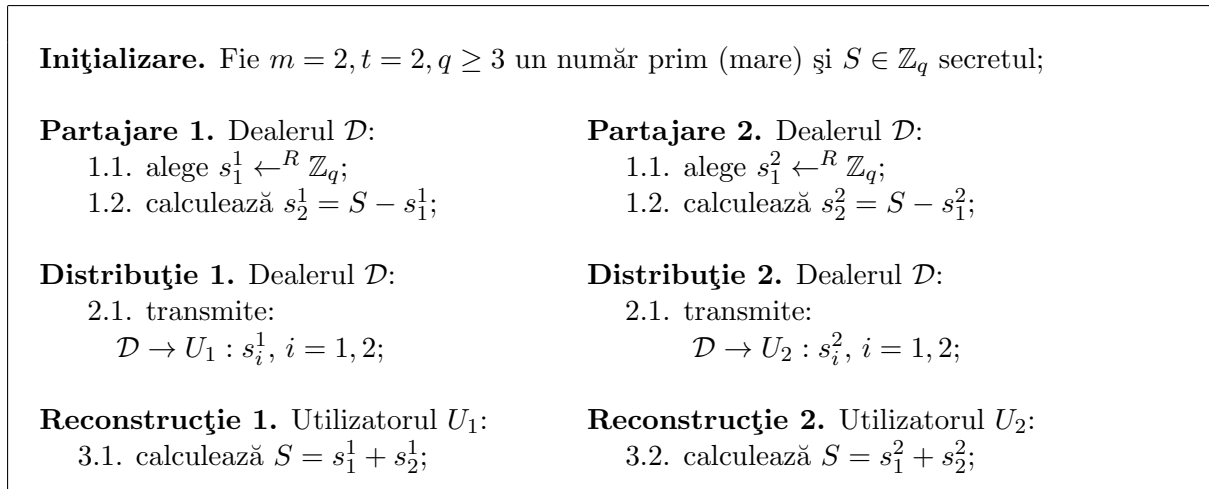


Figura 2.1: O schemă de 2-partajare perfectă bazată pe schema Karnin et al. [20]

Fig.2.1 exemplifică o schemă de 2-partajare perfectă bazată pe schema de partajare prezentată de Karnin et al. [6] și utilizată în cazul general (m partajări) de Sun et al. [23].

2.2 O nouă schemă vizuală de partajare

Am introdus în 2010 o schemă de partajare vizuală care permite împărțirea unei imagini alb-negru în componente color [11]. Fig.2.3 descrie schema în detaliu.

Imaginile sunt reprezentate de mulțimi de pixeli indicați de coordonatele lor ($S(x, y)$ este un pixel al imaginii secrete și $s(x, y)$ un pixel al componentei s) și definiți prin culoare, considerată în modelul RGB. Fig.2.2 ilustrează acest model. Fiecare culoare este codată ca un triplet de octeți care reprezintă proporția de apariție a celor 3 culori primare; listăm cele mai importante valori în Tabelul 2.1.

Fig.2.4 exemplifică o mulțime posibilă de componente în cazul partajării unei imagini de 2x2 pixeli albi, respectiv negri pentru $m = 4$ participanți.

Schema poate fi utilizată numai pentru $m \geq 3$, pentru că cel puțin 3 componente sunt necesare pentru partajarea unui pixel alb (un pixel corespunzător R , unul G și unul B în componente distincte).

Analizăm cantitatea de informație dezvăluită prin cooperarea a t participanți, $1 \leq t \leq m$:

- $t \leq 2$. Cel mult doi participanți nu dețin nici o informație, pentru ca cel puțin 3 componente sunt necesare pentru obținerea unui pixel alb. Așadar, ei pot reconstrui doar o imagine complet neagră.

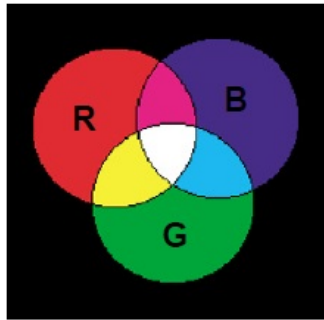


Figura 2.2: Modelul RGB

Tabelul 2.1: Codarea RGB

Culoare	Codare	Culoare	Codare
Alb	$W = (255, 255, 255)$	Negru	$Bl = (0, 0, 0)$
Roșu	$R = (255, 0, 0)$	Galben	$Y = (255, 255, 0)$
Verde	$G = (0, 255, 0)$	Magenta	$M = (255, 0, 255)$
Albastru	$B = (0, 0, 255)$	Cyan	$C = (0, 255, 255)$

Inițializare. Fie S imaginea secretă;

Partajare. Dealerul \mathcal{D} , pentru toți pixelii $S(x, y)$:

- 1.1. dacă $S(x, y) = W$:
 - 1.1.1. alege $s_i(x, y) \leftarrow^R \{R, G, B\}$, $i = 1, \dots, m$;
 - 1.1.2. alege $i_1, i_2, i_3 \leftarrow^R \{1, \dots, m\}$ distincte;
 - 1.1.3. $s_{i_1} \leftarrow R, s_{i_2} \leftarrow G, s_{i_3} \leftarrow B$;
- 1.2. dacă $S(x, y) = Bl$:
 - 1.2.1. alege $X \leftarrow^R \{\{R, G\}, \{R, B\}, \{G, B\}\}$;
 - 1.2.2. alege $s_i(x, y) \leftarrow^R X$, $i = 1, \dots, m$;
 - 1.2.3. alege $i_1, i_2 \leftarrow^R \{1, \dots, m\}$ distincte;
 - 1.2.4. $s_{i_1} \leftarrow^R X, s_{i_2} \leftarrow X \setminus \{s_{i_1}\}$;

Distribuție. Dealerul \mathcal{D} :

- 2.1. transmite (prin canale de comunicație sigure):

$$\mathcal{D} \rightarrow U_i : s_i, i = 1, \dots, m;$$

Reconstrucție. Cei m participanți:

- 3.1. refac imaginea secretă S , unde:
 - 3.1.1. $S(x, y) = W$ dacă există 3 pixeli corespunzători $s_i(x, y)$ distinct colorați, $i = 1, \dots, m$;
 - 3.1.2. $S(x, y) = Bl$ dacă există numai 2 pixeli corespunzători $s_i(x, y)$ distinct colorați $i = 1, \dots, m$.

Figura 2.3: Schema de partajare vizuală Olimid [11]

















4 pixeli albi 	Componenta 1	Componenta 2	Componenta 3	Componenta 4	Imaginea color refăcută 
	RR  RR	GR  RB	BB  BG	RG  GG	
4 pixeli negri 	Componenta 1	Componenta 2	Componenta 3	Componenta 4	Imaginea color refăcută 
	RR  BB 	GB  BB 	RR  GB 	RR  GG 	

Figura 2.4: O posibilă partajare pentru 4 pixeli albi și 4 pixeli negri ($m = 4$) [12]

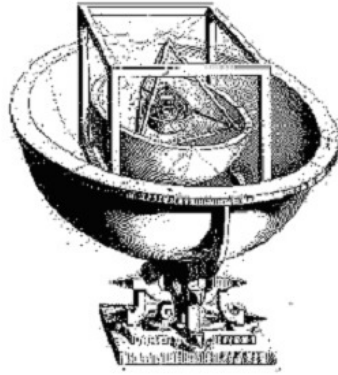
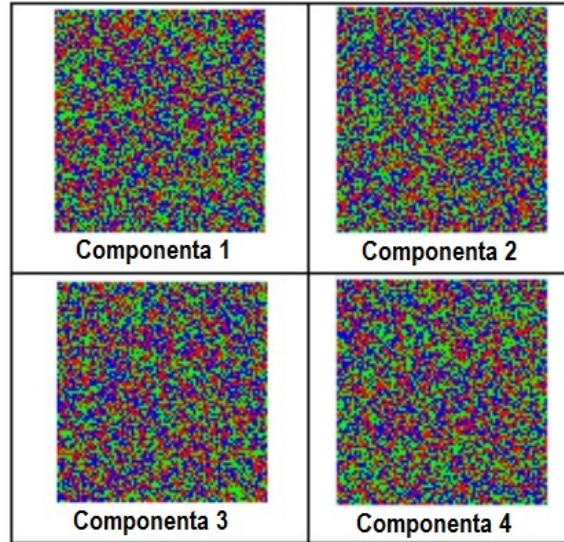


Figura 2.5: Imaginea secretă de test

- $2 < t < m$. Mai mult de 2 participanți obțin informație suplimentară despre secret: un pixel alb obținut prin reconstrucție corespunde cu siguranță unui pixel alb din imaginea inițială; nu se poate afirma însă nimic despre pixelii negri obținuți prin reconstrucție. Probabilitatea de reconstrucție corectă a unui pixel crește cu numărul t de participanți.
- $t = m$. Toți participanții determină imaginea partajată prin punerea în comun a componentelor pe care le dețin.

Am implementat schema propusă în limbajul de programare Python, utilizând IDLE (Integrated DeveLopment Environment 2.6.6) ca mediu de dezvoltare [24]. Un motiv pentru care am ales Python este că pune la dispoziția dezvoltatorilor în variantă gratuită o librărie de lucru cu imagini numită PIL (Python Image Library) care permite: definirea imaginilor în diferite moduri (alb-negru, color conform modelului RBG), manipularea imaginilor la nivel de pixel, a benzilor de culoare (R , G și B), etc. Menționăm că pentru generarea aleatoare a valorilor am utilizat de asemenea modulul *random* existent în Python.

Fig.2.5 indică imaginea inițială. Fig.2.6 exemplifică un set de componente posibile pentru $m = 4$. Fig.2.7 indică posibile reconstrucții din 2 sau 3 componente și imaginea rezultată din utilizarea tuturor componentelor.

Figura 2.6: O posibilă partajare ($m = 4$) [11]


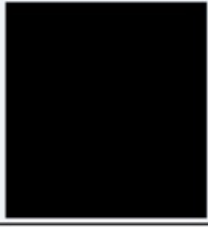




t	Imaginea color reconstruita	Imaginea alb-negru reconstruita
2		
3		
4		

Figura 2.7: Exemple de imagini obținute prin reconstrucție ($m = 4$) [11]

Capitolul 3

SETUP asupra SSS care utilizează generatoare de numere aleatoare

Mecanismul SETUP (Secretly Embedded Trapdoor with Universal Protection) a fost definit de Young și Yung [26]. Acesta reprezintă o tehnică malițioasă realizată de către producătorul unui sistem criptografic care constă în implementarea unui canal care permite scurgerea de informație secretă criptată. Criptarea se realizează folosind cheia publică a atacatorului, acesta fiind deci singurul care are acces la informație.

Implementarea este de tip *cutie neagră* (*black box*): un utilizator are acces numai la valorile de intrare și ieșire ale sistemului, în timp ce algoritmi și memoria internă rămân inaccesibile. Modelul permite astfel atacatorului să schimbe construcția internă, în timp ce utilizatorii nu pot suspecta nici un comportament malițios dacă nu sunt aparent afectate valorile de intrare și ieșire.

Definiția 3.1. (*Secretly Embedded Trapdoor with Universal Protection (SETUP)*) [27]. Fie C un criptosistem de tip cutie neagră cu o specificație publică. Un mecanism de tip SETUP (Secretly Embedded Trapdoor with Universal Protection) este o modificare la nivel algoritmic al lui C cu obținerea lui C' astfel încât:

1. Intrarea lui C' este conformă cu specificațiile publice ale intrării lui C ;
2. C' criptează în mod eficient utilizând cheia publică a atacatorului, memorată în C' ;
3. Cheia privată a atacatorului nu este conținută în C' și este cunoscută numai de către atacator.
4. Ieșirea lui C' este conformă cu specificațiile publice ale ieșirii lui C ;
5. Ieșirile lui C și C' sunt polinomial indistinctibile pentru oricine cu excepția atacatorului;
6. După descoperirea atacului SETUP în implementare (de exemplu prin reverse-engineering) utilizatorii (cu excepția atacatorului) nu pot determina chei anterioare (ideal, nici chei viitoare).

Un criptosistem modificat care implementează SETUP se numește *contaminat* [26].

22 Capitolul 3. SETUP asupra SSS care utilizează generatoare de numere aleatoare

De la introducerea sa, SETUP a fost aplicat diferitelor sisteme de criptare, scheme de semnătură electronică, algoritmi de generare de chei, sisteme de vot electronic. Introducem acum o metodă generală de implementare în cadrul SSS care utilizează suficientă informație aleatoare [14]. Scopul principal este acela de a oferi atacatorului un avantaj semnificativ pentru reconstrucția secretului, fără să necesite acces la o mulțime autorizată de componente.

Atacul SETUP propus devine posibil în următoarele ipoteze:

1. Mecanismul de partajare este implementat ca o cutie neagră care poate stoca într-o memorie non-volatilă informație din mai multe execuții ale SSS;
2. Mecanismul de partajare generează o mulțime de valori aleatoare care sunt ulterior utilizate pentru calculul componentelor (ambele operații se realizează în cadrul cutiei negre);
3. Numărul de valori aleatoare din fiecare componentă este mai mare sau egal cu numărul de componente ale secretului (sau o componentă poate fi considerată un număr aleator);
4. Componentele sunt distribuite prin canale securizate;
5. Atacatorul este întotdeauna unul dintre participanți;
6. Mai multe secrete sunt partajate.

Fig.3.1 descrie algoritmul intern al sistemului contaminat și Fig.3.2 explică reconstrucția secretului așa cum este realizată de atacator. Sistemul de criptare cu cheie publică este ales astfel încât: (1) să fie Ind-CPA sigur; (2) să accepte ca text criptat valid orice valoare din $Sp(s)^u$, unde $u \geq 1$, fixat; (3) $u \in \mathbb{Z}^*$ este minim. Viteza crescută de criptare constituie de asemenea un avantaj.

Teorema 3.1. [14]. Varianta contaminată a SSS satisface proprietatea de indistinctibilitate față de schema originală.

Teorema 3.2. [14] Varianta contaminată a SSS menține securitatea schemei originale pentru oricine cu excepția deținătorului cheii private.

Exemplificăm aplicabilitatea tehnicii propuse asupra schemei Shamir, cea mai cunoscută și utilizată SSS din literatură [14]. Schema este ideală, deci atacatorul nu poate desfășura atacul fără a cunoaște cel puțin o altă componentă ($u > 1$ pentru că nu există un sistem de criptare cu cheie publică Ind-CPA sigur pentru care dimensiunea textului criptat să fie egală cu cea a textului clar). Soluția indicată este optimă: atacatorul necesită cunoașterea unei singure componente suplimentare. Fig.3.3 descrie algoritmul contaminat al schemei Shamir și Fig.3.4 explică procesul de reconstrucție urmat de atacator.

Intrare: m numărul de participanți, S^1, S^2, \dots secretele care se partajează;
Ieșire: $s_1^i, s_2^i, \dots, s_m^i$ componentele corespunzătoare secretului S^i , $i \geq 1$;

- 1: **if** $i==1$ **then**
- 2: $\alpha \leftarrow^R Sp(s)$;
- 3: $s_1^1 = \alpha$;
- 4: $H(ID||s_1^1)$ este stocat în memoria non-volatilă;
- 5: $s_2^1, s_3^1, \dots, s_m^1$ sunt calculate conform cu schema originală, luând în considerare componenta s_1^1 ;
- 6: **else**
- 7: $(s_1^i, \dots, s_u^i) = \text{Enc}(pk, S^i \otimes H(ID||s_1^{i-1}))$, $1 \leq u \leq m$;
- 8: $H(ID||s_1^{i-1})$ este înlocuit în memoria non-volatilă cu $H(ID||s_1^i)$;
- 9: $s_{u+1}^i, s_{u+2}^i, \dots, s_m^i$ sunt calculate conform cu schema originală, luând în considerare componentele s_1^i, \dots, s_u^i .
- 10: **end if**

Figura 3.1: Atac SETUP asupra schemelor de partajare care utilizează valori aleatoare [14]

Pas 1. U_1 memorează componenta sa s_1^{i-1} din execuția anterioară;

Pas 2. U_1 folosește propria componentă s_1^i și s_2^i, \dots, s_u^i pentru a calcula secretul S^i :

$$S^i = \text{Dec}(sk, (s_1^i, \dots, s_u^i)) \otimes (H(ID||s_1^{i-1}))^{-1}.$$

Figura 3.2: Avantajul atacatorului în determinarea secretului în SSS contaminată [14]

Input: m numărul de participanți, S^1, S^2, \dots secretele care se partajează, $q \geq m + 1$ prim;
Output: $s_1^i, s_2^i, \dots, s_m^i$ componentele corespunzătoare secretului S^i , $i \geq 1$;

```

1: if  $i=1$  then
2:    $x_j \leftarrow^R \mathbb{Z}_q$ ,  $j = 1, \dots, m$ ;
3:    $s_1^1 \leftarrow^R \mathbb{Z}_q$ ;
4:   este generat un polinom de grad  $t - 1$  a.î.  $f^1(x) = S^1 + a_1^1 x + \dots + a_{t-1}^1 x^{t-1}$  și
      $f(x_1) = s_1^1$ ;
5:    $s_j^1 = f^1(x_j)$ ,  $j = 2, \dots, m$ ;
6:    $H(ID || s_1^1)$  este stocat în memoria non-volatilă;
7: else
8:    $(s_1^i, s_2^i) = \text{Enc}(pk, S^i \cdot H(ID || s_1^{i-1}))$ ;
9:    $H(ID || s_1^{i-1})$  este înlocuit în memoria non-volatilă cu  $H(ID || s_1^i)$ ;
10:   $a_j^i \leftarrow^R \mathbb{Z}_q$ ,  $j = 3, \dots, t - 1$ ;
11:   $a_1^i$  și  $a_2^i$  sunt calculate ca soluție a sistemului:
           
$$\begin{cases} s_1^i &= f^i(x_1) \\ s_2^i &= f^i(x_2) \end{cases}$$

     unde  $f^i(x) = S^i + a_1^i x + \dots + a_{t-1}^i x^{t-1}$  este un polinom de grad  $t - 1$ ;
12:   $s_j^i = f^i(x_j)$ .
13: end if

```

Figura 3.3: Atac SETUP asupra schemei de partajare Shamir [14]

Pas 1. U_1 memorează componenta sa s_1^{i-1} din execuția anterioară;

Pas 2. U_1 folosește propria componentă s_1^i și s_2^i pentru a calcula secretul S^i :

$$S^i = \text{Dec}(sk, (s_1^i, s_2^i)) \cdot (H(ID || s_1^{i-1}))^{-1}.$$

Figura 3.4: Avantajul atacatorului în determinarea secretului în SSS Shamir contaminată [14]

Partea II

Protocoale de stabilire a cheilor de grup bazate pe scheme de partajare a secretelor

Capitolul 4

Introducere în protocoale de stabilire a cheilor de grup

Definiția 4.1. (*Protocoale de stabilire a cheilor de grup (GKE¹)*) [8]. Un *protocol de stabilire a cheilor de grup (GKE)* este un proces sau un protocol prin care un secret devine comun mai multor părți, în scopul unei utilizări criptografice ulterioare.

Scopul principal al unui protocol GKE este acela de a stabili o cheie comună între membrii autorizați ai unui grup, fără a scurge informație altor entități. Participanții *autorizați* se mai numesc *calificați*, *legitimi* sau *privilegiați*. Un protocol se execută de mai multe ori, fiecare execuție reprezentând o *sesiune*, identificată în mod unic de un identificator (**sid**²). Numim *cheie de sesiune* un secret comun rezultat în urma unei execuții a protocolului. Aceasta persistă pentru o scurtă perioadă de timp, o prezumție naturală în criptografie. Pentru a deveni eligibil să ia parte la o sesiune, un utilizator trebuie ca în prealabil să se înregistreze în cadrul grupului, operație în urma căreia dobândește o *cheie de lungă durată* pe care o va utiliza pe parcursul derulării protocolului.

În general, protocoalele GKE constau în mai multe faze:

1. *Inițializarea.* Definește mediul de desfășurare al protocolului: parametrii, spațiul cheilor posibile, alte premise.
2. *Înregistrarea.* În funcție de scenariu, după înregistrare, un utilizator poate partaja o cheie secretă (sau o parolă) cu o autoritate de încredere (KGC³) sau poate genera o pereche cheie publică - cheie privată pentru semnături ulterioare.
3. *Execuția.* Descrie algoritmul criptografic, inclusiv operațiile executate și mesajele transmise. De obicei constă în mai multe runde de comunicație între participanți.
4. *Calculul cheii de grup.* Explicitează algoritmi necesari pentru obținerea cheii din cunoștințele dobândite după faza de execuție. Este câteodată integrată ca o rundă în faza de execuție.

¹Group Key Establishment.

²Session id.

³Key Generation Center.

5. *Confirmarea cheii de grup.* Asigură că toate entitățile autorizate dețin într-adevăr cheia privată comună și nimeni în afara lor nu are acces la aceasta. Este o fază opțională care în general apare din motive de securitate.

Protocoalele GKE se împart în două clase: *protocoale de transfer (GKT⁴)* și *protocoale de punere de acord (GKA⁵)*.

Definiția 4.2. (Protocoale de transfer al cheilor de grup (GKT)) [8]. Un *protocol de transfer al cheilor de grup (GKT)* este un protocol de stabilire a cheilor de grup în care o entitate generează (sau obține prin alte mijloace) o cheie și o transferă prin canale sigure celorlalte entități autorizate.

Definiția 4.3. (Protocoale de punere de acord a cheilor de grup (GKA)) [8]. Un *protocol de punere de acord a cheilor de grup (GKA)* este un protocol de stabilire a cheilor de grup în care cheia secretă comună se obține în urma contribuției tuturor entităților autorizate astfel încât nici una să nu poată determina valoarea acesteia.

Schemele de partajare a secretelor se utilizează în construcția protocoalelor de stabilire a cheilor de grup, permițând construcții eficiente: utilizatorii pot comunica numai prin canale de tip broadcast, calculul cheii se poate reduce la simple ecuații liniare, numărul de runde poate rămâne constant indiferent de numărul de participanți. În plus, oferă o modalitate simplă de a diferenția membrii în funcție de puterea deținută în cadrul grupului, permit delegarea sarcinilor, detecția trișorilor și o dimensionare facilă a grupului în funcție de un prag [21].

Multiple construcții GKE bazate pe scheme de partajare a secretelor există în literatură, unele dintre acestea fiind prezentate în varianta completă a tezei. În continuare ne vom referi strict la cele legate de contribuțiile personale.

⁴Group Key Transfer.

⁵Group Key Agreement.

Capitolul 5

Protocele GKE informal sigure bazate pe SSS

Tabelul 5.1 sumarizează rezultatele la care se face referire pe parcursul capitolului.

Tabelul 5.1: Atacuri și îmbunătățiri aplicate protocelelor GKE informal sigure

Protocolul original	Atacuri și Îmbunătățiri
Harn și Lin (Jun 2010) [4]	Nam et al. (Dec 2011) [9] Yuan et al. (Sep 2013) [29], Olimid [17]
Hsu et al. (Jan 2012) [5]	Olimid [19]
Sun et al. (Mar 2012) [23]	Olimid (Mar 2013) [18], Kim et al. (May 2013) [7]
Yuan et al. (Jan 2013) [28]	Olimid (Jul 2013) [10], [15]

5.1 Harn și Lin (2010)

Harn și Lin au introdus în 2010 un protocol GKT [4] bazat pe schema de partajare a lui Shamir [22]. Nam et al. arată vulnerabilitatea protocolului la un atac prin reluare [9]. Recent, Yuan et al. introduc un atac de tip man-in-the-middle și propun o modalitate de îmbunătățire a protocolului pentru a rezista la atac [29]. Fig.5.1 descrie varianta Yuan et al. a protocolului lui Harn și Lin.

Contrar a ceea ce afirmă autorii, varianta propusă rămâne vulnerabilă la un atac de tip man-in-the-middle, prezentat în detaliu în Fig.5.2 [17].

5.2 Hsu et al. (2012)

Hsu et al. au introdus în anul 2012 un protocol GKT bazat pe perspectiva non-tradițională a schemelor de m -partajare prezentată în Secțiunea 2.1 [5]. Fig 5.3 descrie protocolul în detaliu.

Inițializare. *KGC* selectează 2 numere prime sigure p și q (i.e. $p' = \frac{p-1}{2}$ și $q' = \frac{q-1}{2}$ sunt de asemenea prime) și calculează $n = pq$;

Apoi, alege $e \in \mathbb{Z}_n^*$ a.î. $(e, \Phi(n)) = 1$ și calculează $d \in \mathbb{Z}_n^*$ a.î. $ed = 1 \pmod{\Phi(n)}$;

Înregistrare. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește un secret de lungă durată $(x_i, y_i) \in \mathbb{Z}_n^* \times \mathbb{Z}_n^*$ cu *KGC* și primește cheia sa publică (e, n) ;

Runda 1. Utilizatorul U_1 :

1.1. transmite:

$$U_1 \rightarrow KGC : \{U_1, \dots, U_t\}$$

Runda 2. *KGC*:

2.1. calculează $v = h(U_1, \dots, U_t)^d \pmod{n}$;

2.2. transmite:

$$KGC \rightarrow^* : (\{U_1, \dots, U_t\}, v)$$

Runda 3. Fiecare utilizator $U_i, i = 1, \dots, t$:

3.1. verifică dacă $h(U_1, \dots, U_t) = v^e \pmod{n}$;

Dacă egalitatea nu se satisface, abandonează.

3.2. alege $R_i \leftarrow^R \mathbb{Z}_n^*$;

3.3. transmite:

$$U_i \rightarrow KGC : R_i$$

Runda 4. *KGC*:

4.1. alege o cheie $k \leftarrow^R \mathbb{Z}_n^*$;

4.2. generează polinomul $f(x)$ de grad t care trece prin $t + 1$ puncte $(0, k)$,

$$(x_1, y_1 \oplus R_1), \dots, (x_t, y_t \oplus R_t);$$

4.3. calculează t puncte adiționale P_1, \dots, P_t ale lui $f(x)$;

4.4. calculează mesajul de autentificare $Auth = h(k, U_1, \dots, U_t, R_1, \dots, R_t, P_1, \dots, P_t)$;

4.5. transmite:

$$KGC \rightarrow^* : (P_1, \dots, P_t, R_1, \dots, R_t, Auth)$$

Calculul cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

5.1. calculează cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și $(x_i, y_i \oplus R_i)$;

5.2. verifică dacă $Auth = h(k, U_1, \dots, U_t, R_1, \dots, R_t, P_1, \dots, P_t)$;

Dacă egalitatea nu se satisface, abandonează.

Figura 5.1: Versiunea Yuan et al. a protocolului de transfer de chei Harn și Lin [29]

Pas 1. U_a nu intervine în Runda 1 a unei sesiuni legitime, dar interceptează răspunsul $(\{U_1, \dots, U_t\}, v)$ transmis în Runda 2 și previne ca acesta să ajungă la participanți.

Pas 2. U_a joacă rolul fiecărui participant la sesiune și trimite o valoare R'_j în numele lui U_j , $j = 1, \dots, t$:

$$U_a(U_j) \rightarrow KGC : R'_j$$

Pas 3. U_a interceptează mesajul $(R'_1, \dots, R'_t, P'_1, \dots, P'_t, Auth')$ transmis în Runda 4 și previne ca acesta să ajungă la utilizatori.

Pas 4. U_a inițiază o nouă sesiune:

$$U_a \rightarrow KGC : \{U_1, \dots, U_{i-1}, U_a, U_{i+1}, \dots, U_t\}$$

Pas 5. U_a interceptează mesajul $(\{U_1, \dots, U_{i-1}, U_a, U_{i+1}, \dots, U_t\}, v')$ transmis în Runda 2 a noii sesiuni și previne ca acesta să ajungă la utilizatori;

Pas 6. U_a transmite mesajul de la Pasul 1 tuturor membrilor, cu excepția lui U_i ; Utilizatorii U_j , $j = 1, \dots, t$, $j \neq i$, verifică $h(U_1, \dots, U_t) = v^e \pmod{n}$ (și deci nu poate detecta atacul);

Pas 7. U_a permite mesajelor R_j , $j = 1, \dots, t$, $j \neq i$ să ajungă la KGC , alege $R_a \leftarrow^R \mathbb{Z}_n^*$ și transmite către KGC :

$$U_a \rightarrow KGC : R_a$$

Pas 8. U_a interceptează mesajul $(R_1, \dots, R_{i-1}, R_a, R_{i+1}, \dots, R_t, P_1, \dots, P_t, Auth)$ în Runda 4, calculează cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și $(x_a, y_a \oplus R_a)$, apoi transmite mesajul participanților (ca provenind de la KGC).

Pas 9. Utilizatorii U_j , $j = 1, \dots, t$, $j \neq i$, determină cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și $(x_j, y_j \oplus R_j)$, verifică dacă $Auth = h(k, U_1, \dots, U_t, R_1, \dots, R_{i-1}, R_a, R_{i+1}, \dots, R_t, P_1, \dots, P_t)$ se satisface și acceptă k drept cheie corectă de grup.

Figura 5.2: Atac man-in-the-middle asupra versiunii Yuan et al. a protocolului Harn și Lin [17]

Protocolul Hsu et al. este vulnerabil la un atac activ din interior, prezentat în Fig.5.4 [19]. Atacul este cauzat de o slăbiciune de autentificare: cheia de grup nu este autentificată ca provenind de la inițiatorul U_1 . Acest fapt îi permite adversarului să joace rolul lui U_1 și să trimită un mesaj modificat, dar aparent autentic. Propunem în Fig.5.5 o variantă îmbunătățită a protocolului care rezistă atacului.

Inițializare. Fie G un grup ciclic multiplicativ de ordin p , cu g generator, unde p este un număr prim sigur mare (i.e. $p' = \frac{p-1}{2}$ este prim);

Înregistrare. Fiecare utilizator U_i , $i = 1, \dots, m$ deține o pereche cheie publică - cheie privată (pk_i, sk_i) a.î. $pk_i = g^{sk_i}$ în G .

Runda 1. Utilizatorul U_1 :

- 1.1. alege $r_1 \leftarrow^R \mathbb{Z}_p^*$;
- 1.2. transmite o cerere de generare a cheii:

$$U_1 \rightarrow^*: (\{U_1, \dots, U_t\}, r_1, pk_1)$$

Runda 2. Fiecare utilizator U_i , $i = 2, \dots, t$:

- 2.1. alege $r_i \leftarrow^R \mathbb{Z}_p^*$;
- 2.2. calculează $S_i = pk_1^{sk_i r_i r_1}$ un secret comun cu U_1 și $Auth_i = h(S_i, r_1)$;
- 2.3. transmite:

$$U_i \rightarrow^*: (r_i, pk_i, Auth_i)$$

Runda 3. Utilizatorul U_1 :

- 3.1. calculează $S_i = pk_i^{sk_1 r_i r_1}$, $i = 2, \dots, t$;
- 3.2. verifică dacă $Auth_i = h(S_i, r_1)$, $i = 2, \dots, t$;
 Dacă cel puțin o egalitate nu se satisface, abandonează;
- 3.3. alege cheia de grup $k \leftarrow^R \mathbb{Z}_p^*$, împarte fiecare secret S_i în două părți $S_i = x_i || y_i$ și calculează $t - 1$ valori $K_i = k - T_i$, unde $T_i = (y_i v(x_i), r)$ este produsul scalar al vectorilor $y_i v(x_i) = y_i \sum_{j=1}^t x_i^{j-1} e_j$ ($e_j = (0, \dots, 1, \dots, 0)$ cu 1 pe poziția j), $i = 2, \dots, t$ și $r = (r_1, \dots, r_t)$;
- 3.4. calculează $Auth = h(k, U_1, \dots, U_t, r_1, \dots, r_t, K_2, \dots, K_t)$;
- 3.5. transmite:

$$U_1 \rightarrow^*: (Auth, K_2, \dots, K_t)$$

Calculul Cheii. Fiecare utilizator U_i , $i = 2, \dots, t$:

- 4.1. calculează produsul scalar $T_i = (y_i v(x_i), r)$, determină cheia de grup $k = T_i + K_i$;
- 4.2. verifică dacă $Auth = h(k, U_1, \dots, U_t, r_1, \dots, r_t, K_2, \dots, K_t)$;
 Dacă egalitatea nu se satisface, abandonează.

Figura 5.3: Versiunea originală a protocolului de transfer de chei Hsu et al. [5]

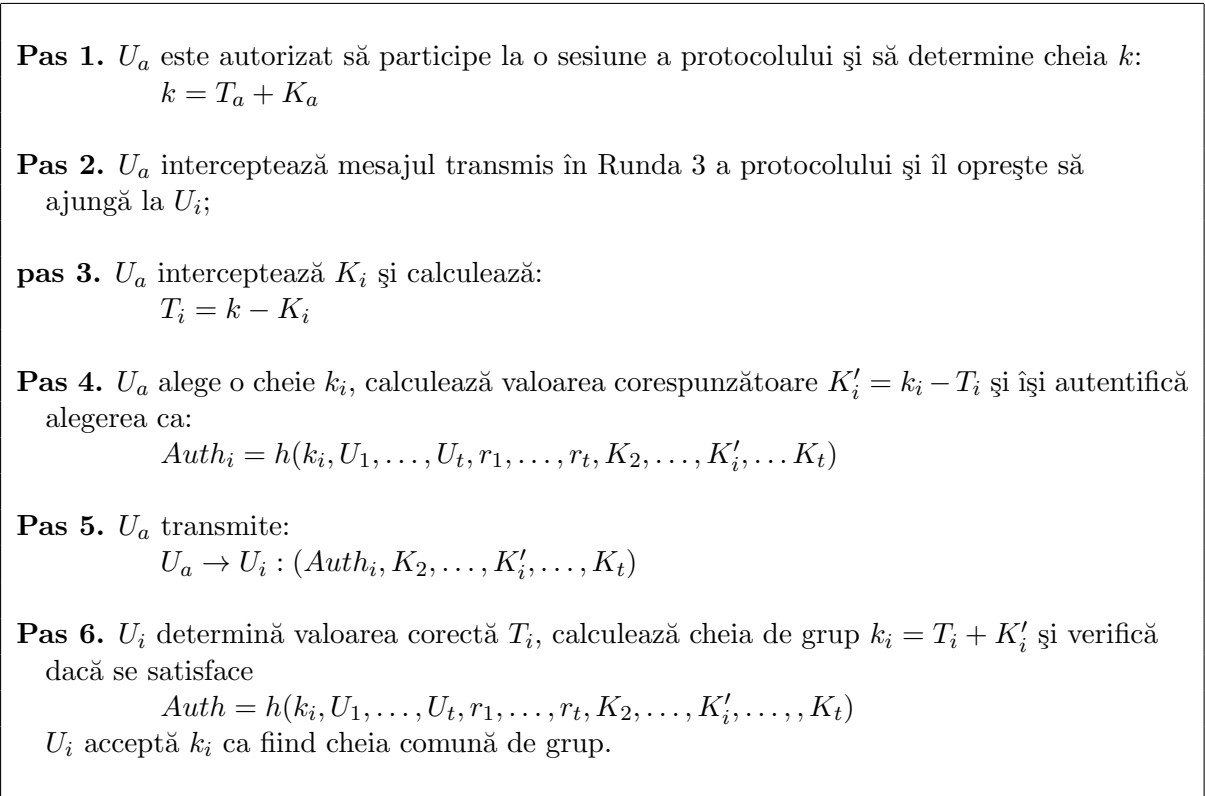


Figura 5.4: Atac din interior asupra versiunii originale a protocolului Hsu et al. [19]

5.3 Sun et al. (2012)

Sun et al. au introdus în anul 2012 un protocol GKT bazat pe perspectiva non-tradițională a schemelor de m -partajare prezentată în Secțiunea 2.1 [23]. Fig.5.6 descrie protocolul în detaliu.

Protocolul Sun et al.'s este vulnerabil la un atac din interior, prezentat în Fig.5.7 și la un atac de tip *cheie cunoscută*¹, detaliat în Fig.5.8 [18]. Ambele atacuri sunt cauzate de faptul că valorile $g^{s'_i}$, $i = 1, \dots, m$ se mențin pentru mai multe sesiuni. Propunem în Fig.5.9 o modalitate de eliminare a acestei probleme [18] inspirată din lucrarea lui Pieprzyk și Li [21].

5.4 Yuan et al. (2013)

Yuan et al. au introdus în anul 2013 un protocol GKT bazat pe schema de partajare Shamir [22] în care utilizatorii, ca urmare a înregistrării în grup, dețin parole de lungă durată [28]. Fig.5.10 descrie protocolul în detaliu.

Protocolul este vulnerabil la un *atac prin reluare*², prezentat în Fig.5.11 [10], [15]. Atacul

¹Known key attack.

²Replay attack.

Inițializare. Fie G un grup ciclic multiplicativ de ordin p , cu g generator, unde p este un număr prim sigur mare (i.e. $p' = \frac{p-1}{2}$ este prim);

Înregistrare. Fiecare utilizator U_i , $i = 1, \dots, m$ deține o pereche cheie publică - cheie privată (pk_i, sk_i) a.î. $pk_i = g^{sk_i}$ în G .

Runda 1. Utilizatorul U_1 :

- 1.1. alege $r_1 \leftarrow^R \mathbb{Z}_p^*$;
- 1.2. transmite o cerere de generare a cheii:

$$U_1 \rightarrow^*: (\{U_1, \dots, U_t\}, r_1, pk_1)$$

Runda 2. Fiecare utilizator U_i , $i = 2, \dots, t$:

- 2.1. alege $r_i \leftarrow^R \mathbb{Z}_p^*$;
- 2.2. calculează $S_i = pk_1^{sk_i r_i r_1}$ un secret comun cu U_1 și $Auth_i = h(S_i, r_1)$;
- 2.3. transmite:

$$U_i \rightarrow^*: (r_i, pk_i, Auth_i)$$

Runda 3. Utilizatorul U_1 :

- 3.1. calculează $S_i = pk_i^{sk_1 r_i r_1}$, $i = 2, \dots, t$;
- 3.2. verifică dacă $Auth_i = h(S_i, r_1)$, $i = 2, \dots, t$;
 Dacă cel puțin o egalitate nu se satisface, abandonează;
- 3.3. alege cheia de grup $k \leftarrow^R \mathbb{Z}_p^*$, împarte fiecare secret S_i în două părți $S_i = x_i || y_i$ și calculează $t - 1$ valori $K_i = k - T_i$, unde $T_i = (y_i v(x_i), r)$ este produsul scalar al vectorilor $y_i v(x_i) = y_i \sum_{j=1}^t x_i^{j-1} e_j$ ($e_j = (0, \dots, 1, \dots, 0)$ cu 1 pe poziția j), $i = 2, \dots, t$ și $r = (r_1, \dots, r_t)$;
- 3.4. calculează $Auth = \Sigma.\text{Sign}_{U_1}(h(k, U_1, \dots, U_t, r_1, \dots, r_t, K_2, \dots, K_t))$;
- 3.5. transmite:

$$U_1 \rightarrow^*: (Auth, K_2, \dots, K_t)$$

Calculul Cheii. Fiecare utilizator U_i , $i = 2, \dots, t$:

- 4.1. calculează produsul scalar $T_i = (y_i v(x_i), r)$, determină cheia de grup $k = T_i + K_i$;
- 4.2. verifică dacă $\Sigma.\text{Verify}_{U_1}(h(k, U_1, \dots, U_t, r_1, \dots, r_t, K_2, \dots, K_t), Auth) = 1$.
 Dacă egalitatea nu se satisface, abandonează.

Figura 5.5: Versiunea îmbunătățită a protocolului de transfer de chei Hsu et al. [19]

Inițializare. *KGC* alege un grup ciclic multiplicativ G de ordin prim p cu g generator;

Înregistrare. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește un secret de lungă durată $s'_i \in G$ comun cu *KGC*;

Runda 1. Utilizatorul U_1 :

1.1. transmite o cerere de generare a cheii:

$$U_1 \rightarrow KGC : \{U_1, \dots, U_t\}$$

Runda 2. *KGC*:

2.1. transmite:

$$KGC \rightarrow^* : \{U_1, \dots, U_t\}$$

Runda 3. Fiecare utilizator $U_i, i = 1, \dots, t$:

3.1. alege $r_i \leftarrow^R \mathbb{Z}_p^*$;

3.2. transmite:

$$U_i \rightarrow^* : r_i$$

Runda 4. *KGC*:

4.1. alege $S \leftarrow^R G$;

4.2. partajează S în 2 componente de t ori a.î. $S = s_i + s'_i, i = 1, \dots, t$;

4.3. calculează $k = g^S, M_i = (g^{s_i+r_i}, U_i, h(U_i, g^{s_i+r_i}, s'_i, r_i)), i = 1, \dots, t$ și $Auth = h(k, g^{s_1+r_1}, \dots, g^{s_t+r_t}, U_1, \dots, U_t, r_1, \dots, r_t)$;

4.4. transmite:

$$KGC \rightarrow^* : (M_1, \dots, M_t, Auth)$$

Calculul cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

5.1. verifică dacă $h(U_i, g^{s_i+r_i}, s'_i, r_i)$ este egal cu valoarea corespunzătoare din M_i ;
Dacă egalitatea nu se satisface, abandonează;

5.2. calculează cheia de grup $k = g^{s'_i} g^{s_i+r_i} (g^{r_i})^{-1}$;

5.3. verifică dacă $Auth = h(k, g^{s_1+r_1}, \dots, g^{s_t+r_t}, U_1, \dots, U_t, r_1, \dots, r_t)$;

Dacă egalitatea nu se satisface, abandonează;

5.4. calculează $h_i = h(s'_i, k, U_1, \dots, U_t, r_1, \dots, r_t)$

5.4. transmite:

$$U_i \rightarrow KGC : h_i$$

Confirmarea cheii. *KGC*:

6.1. verifică dacă $h_i = h(s'_i, k, U_1, \dots, U_t, r_1, \dots, r_t), i = 1, \dots, t$, certificând că toți utilizatorii dețin aceeași cheie.

Figura 5.6: Versiunea originală a protocolului de transfer de chei Sun et al. [23]

Pas 1. U_a este autorizat să participe la o sesiune (s_1) și să determine cheia $k_{(s_1)}$:

$$k_{(s_1)} = \frac{g^{s'_a} \cdot g^{s_a(s_1) + r_a(s_1)}}{g^{r_a(s_1)}}$$

Pas 2. Cum $r_{i(s_1)}$ și $g^{s_{i(s_1)} + r_{i(s_1)}}$ sunt transmise în clar în Rundele 3 și 4 ale protocolului, U_a poate calcula $g^{s'_i}$ pentru orice $U_i \in \mathcal{U}_{(s_1)}$:

$$g^{s'_i} = \frac{k_{(s_1)} \cdot g^{r_{i(s_1)}}}{g^{s_{i(s_1)} + r_{i(s_1)}}}$$

Pas 3. U_a interceptează $r_{j(s_2)}$ și $g^{s_{j(s_2)} + r_{j(s_2)}}$ într-o sesiune (s_2) a.î. $U_a \notin \mathcal{U}_{(s_2)}$ și determină, pentru orice $U_j \in \mathcal{U}_{(s_2)}$:

$$g^{s_{j(s_2)}} = \frac{g^{s_{j(s_2)} + r_{j(s_2)}}}{g^{r_{j(s_2)}}}$$

Pas 4. Dacă există $U_b \in \mathcal{U}_{(s_1)} \cap \mathcal{U}_{(s_2)}$, atunci U_a calculează cheia $k_{(s_2)}$ a sesiunii (s_2) ca:

$$k_{(s_2)} = g^{s'_b} \cdot g^{s_{b(s_2)}} = g^{s'_b + s_{b(s_2)}}.$$

Figura 5.7: Atac din interior asupra versiunii originale a protocolului Sun et al. [18]

Pas 1. Cum $r_{i(s_1)}$ și $g^{s_{i(s_1)} + r_{i(s_1)}}$ sunt transmise în clar în Rundele 3 și 4 ale protocolului, U_a poate calcula $g^{s'_i}$ pentru orice $U_i \in \mathcal{U}_{(s_1)}$ sub premisa că a reușit prin diferite mijloace să determine cheia de sesiune $k_{(s_1)}$:

$$g^{s'_i} = \frac{k_{(s_1)} \cdot g^{r_{i(s_1)}}}{g^{s_{i(s_1)} + r_{i(s_1)}}}$$

Pas 2. U_a interceptează $r_{j(s_2)}$ și $g^{s_{j(s_2)} + r_{j(s_2)}}$ într-o sesiune (s_2), a.î. ($s_2 \neq s_1$), $U_a \notin \mathcal{U}_{(s_2)}$ și determină, pentru orice $U_j \in \mathcal{U}_{(s_2)}$:

$$g^{s_{j(s_2)}} = \frac{g^{s_{j(s_2)} + r_{j(s_2)}}}{g^{r_{j(s_2)}}}$$

Pas 3. Dacă există $U_b \in \mathcal{U}_{(s_1)} \cap \mathcal{U}_{(s_2)}$, atunci U_a determină cheia $k_{(s_2)}$ a sesiunii (s_2) ca:

$$k_{(s_2)} = g^{s'_b} \cdot g^{s_{b(s_2)}} = g^{s'_b + s_{b(s_2)}}.$$

Figura 5.8: Atac cu cheie cunoscută asupra versiunii originale a protocolului Sun et al. [18]

este posibil pentru că KGC nu poate detecta reutilizarea mesajelor. Propunem în Fig.5.12 o modalitate de eliminare a acestei slăbiciuni [10], [15] analogă celei introduse de Nam et al. [9] pentru protocolul similar al lui Harn și Lin [4].

Versiunea propusă rămâne însă vulnerabilă la un atac din interior, prezentat în Fig.5.13 [15]. Acesta este cauzat de faptul că pw_{iy} poate fi exprimat ca un polinom cu coeficienți cunoscuți în pw_{ix} , ceea ce permite atacatorului să obțină un sistem de ecuații în singura

Inițializare. *KGC* alege un grup ciclic multiplicativ G de ordin prim p cu g generator;

Înregistrare. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește un secret de lungă durată $s'_i \in G$ comun cu *KGC*;

Runda 1. Utilizatorul U_1 :

1.1. transmite o cerere de generare a cheii:

$$U_1 \rightarrow KGC : \{U_1, \dots, U_t\}$$

Runda 2. *KGC*:

2.1. transmite:

$$KGC \rightarrow^* : \{U_1, \dots, U_t\}$$

Runda 3. Fiecare utilizator $U_i, i = 1, \dots, t$:

3.1. alege $r_i \leftarrow^R \mathbb{Z}_p^*$;

3.2. transmite:

$$U_i \rightarrow^* : r_i$$

Runda 4. *KGC*:

4.1. alege $S \leftarrow^R G$ și $\alpha \leftarrow^R G$;

4.2. partajează S în 2 componente de t ori a.î. $S = s_i + s'_i, i = 1, \dots, t$;

4.3. calculează $k = \alpha^S, M_i = (\alpha^{s_i+r_i}, U_i, h(U_i, \alpha^{s_i+r_i}, s'_i, r_i, \alpha)), i = 1, \dots, t$ și $Auth = h(k, \alpha^{s_1+r_1}, \dots, \alpha^{s_t+r_t}, U_1, \dots, U_t, r_1, \dots, r_t, \alpha)$;

4.4. transmite:

$$KGC \rightarrow^* : (M_1, \dots, M_t, Auth, \alpha)$$

Calculul cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

5.1. verifică dacă $h(U_i, g^{s_i+r_i}, s'_i, r_i, \alpha)$ este egal cu valoarea corespunzătoare din M_i ;
Dacă egalitatea nu se satisface, abandonează;

5.2. calculează cheia de grup $k = \alpha^{s'_i} \alpha^{s_i+r_i} (\alpha^{r_i})^{-1}$;

5.3. verifică dacă $Auth = h(k, \alpha^{s_1+r_1}, \dots, \alpha^{s_t+r_t}, U_1, \dots, U_t, r_1, \dots, r_t, \alpha)$;

Dacă egalitatea nu se satisface, abandonează;

5.4. calculează $h_i = h(s'_i, k, U_1, \dots, U_t, r_1, \dots, r_t, \alpha)$

5.4. transmite:

$$U_i \rightarrow KGC : h_i$$

Confirmarea cheii. *KGC*:

6.1. verifică dacă $h_i = h(s'_i, k, U_1, \dots, U_t, r_1, \dots, r_t, \alpha), i = 1, \dots, t$, certificând că toți utilizatorii dețin aceeași cheie.

Figura 5.9: Versiunea îmbunătățită a protocolului de transfer de chei Sun et al. [18]

Inițializare. *KGC* alege 2 numere prime mari p și q și calculează $n = pq$;

Înregistrarea. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește o parolă de lungă durată $pw_i = pw_{ix} || pw_{iy}$ cu *KGC*;

Runda 1. Utilizatorul U_1 :

- 1.1. alege $k_1 \leftarrow^R \mathbb{Z}_n$;
- 1.2. calculează $K_1 = pw_{1x} + k_1$ și $M_1 = h_1(U_1, \dots, U_t, k_1)$;
- 1.3. transmite o cerere de generare a cheii:

$$U_1 \rightarrow KGC : (U_1, \{U_1, \dots, U_t\}, K_1, M_1)$$

Runda 2. *KGC*:

- 2.1. calculează $k_1 = K_1 - pw_{1x}$;
- 2.2. verifică dacă $M_1 = h_1(U_1, \dots, U_t, k_1)$;
 Dacă egalitatea nu se satisface, abandonează;
- 2.3. transmite:

$$KGC \rightarrow^* : \{U_1, \dots, U_t\}$$

Runda 3. Fiecare utilizator $U_i, i = 2, \dots, t$:

- 3.1. alege $k_i \leftarrow^R \mathbb{Z}_n$;
- 3.2. calculează $K_i = pw_{ix} + k_i$ și $M_i = h_1(U_1, \dots, U_t, k_i)$;
- 3.3. transmite:

$$U_i \rightarrow KGC : (U_i, \{U_1, \dots, U_t\}, K_i, M_i)$$

Runda 4. *KGC*:

- 4.1. calculează $k_i = K_i - pw_{ix}, i=2, \dots, t$;
- 4.2. verifică dacă $M_i = h_1(U_1, \dots, U_t, k_i), i=2, \dots, t$;
 Dacă egalitatea nu se satisface, abandonează;
- 4.3. alege 2 numere aleatoare x_{ta} și y_{ta} de lungime egală cu pw_{ix} și pw_{iy} ;
- 4.4. generează polinomul $f(x)$ de grad t care trece prin $t + 1$ puncte $(x_{ta}, y_{ta}), (pw_{1x}, pw_{1y} + k_1), \dots, (pw_{tx}, pw_{ty} + k_t)$;
- 4.5. calculează t puncte adiționale P_1, \dots, P_t ale lui $f(x)$;
- 4.6. calculează mesajul de verificare $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i), i = 1, \dots, t$;
- 4.7. transmite, $i = 1, \dots, t$:

$$KGC \rightarrow U_i : (P_1, \dots, P_t, V_i)$$

Calculul cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

- 5.1. verifică dacă $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i)$;
 Dacă egalitatea nu se satisface, abandonează;
- 5.2. calculează cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și $(pw_{ix}, pw_{iy} + k_i)$.

Figura 5.10: Versiunea originală a protocolului de transfer de chei Yuan et al. [28]

- Pas 1.** U_a inițializează o sesiune (s_1) cerând stabilirea unei chei comune cu U_i ;
- Pas 2.** U_a interceptează $(U_i, \{U_i, U_a\}, K_i, M_i)$ în Runda 3 a protocolului;
- Pas 3.** U_a inițializează o altă sesiune (s_2) cerând stabilirea unei chei comune cu U_i și utilizează aceeași valoare k_a pentru ambele sesiuni (s_1) și (s_2);
- Pas 4.** U_a joacă rolul lui U_i în sesiunea (s_2) și trimite în Runda 3 mesajul $(U_i, \{U_i, U_a\}, K_i, M_i)$ pe care l-a interceptat în pasul 2;
- Pas 5.** U_a este un utilizator autorizat pentru ambele sesiuni, așa încât determină

$$f(x)_{(s_j)} = a_{(s_j)}x^2 + b_{(s_j)}x + c_{(s_j)}, j = 1, 2$$
- Pas 6.** Cum $(pw_{ax}, pw_{ay} + k_a)$ și $(pw_{ix}, pw_{iy} + k_i)$ sunt puncte valide ale lui $f(x)_{(s_j)}$, U_a știe că $f(pw_{ax})_{(s_1)} = f(pw_{ax})_{(s_2)} = pw_{ay} + k_a$ și $f(pw_{ix})_{(s_1)} = f(pw_{ix})_{(s_2)} = pw_{iy} + k_i$; deci pw_{ax} și pw_{ix} sunt rădăcini pentru:

$$(a_{(s_1)} - a_{(s_2)})x^2 + (b_{(s_1)} - b_{(s_2)})x + (c_{(s_1)} - c_{(s_2)}) = 0$$
- Pas 7.** U_a determină parola de lungă durată a lui U_i ca:

$$pw_{ix} = pw_{ax}^{-1}(a_{(s_1)} - a_{(s_2)})^{-1}(c_{(s_1)} - c_{(s_2)}).$$

$$pw_{iy} = f(pw_{ix})_{(s_j)} - K_{i(s_j)} + pw_{ix}, \text{ for any } j = 1, 2$$

Figura 5.11: Atac prin reluare asupra versiunii originale a protocolului Yuan et al. [10]

necunoscută pw_{ix} . Propunem în Fig.5.14 o variantă îmbunătățită a protocolului care rezistă acestui atac [10].

Inițializare. *KGC* alege 2 numere prime mari p și q și calculează $n = pq$;

Înregistrarea. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește o parolă de lungă durată $pw_i = pw_{ix} || pw_{iy}$ cu *KGC*;

Runda 1. Utilizatorul U_1 :

1.1. transmite o cerere de generare a cheii:

$$U_1 \rightarrow KGC : \{U_1, \dots, U_t\}$$

Runda 2. *KGC*:

2.1. alege $k_0 \leftarrow^R \mathbb{Z}_n$;

2.2. transmite:

$$KGC \rightarrow^* : (\{U_1, \dots, U_t\}, k_0)$$

Runda 3. Fiecare utilizator $U_i, i = 1, \dots, t$:

3.1. alege $k_i \leftarrow^R \mathbb{Z}_n$;

3.2. calculează $K_i = pw_{ix} + k_i$ și $M_i = h_1(U_1, \dots, U_t, k_i, k_0)$;

3.3. transmite:

$$U_i \rightarrow KGC : (U_i, \{U_1, \dots, U_t\}, K_i, M_i)$$

Runda 4. *KGC*:

4.1. calculează $k_i = K_i - pw_{ix}, i=1, \dots, t$;

4.2. verifică dacă $M_i = h_1(U_1, \dots, U_t, k_i, k_0), i=1, \dots, t$;

Dacă egalitatea nu se satisface, abandonează;

4.3. alege 2 numere aleatoare x_{ta} și y_{ta} de lungime egală cu pw_{ix} și pw_{iy} ;

4.4. generează polinomul $f(x)$ de grad t care trece prin $t + 1$ puncte $(x_{ta}, y_{ta}), (pw_{1x}, pw_{1y} + k_1), \dots, (pw_{tx}, pw_{ty} + k_t)$;

4.5. calculează t puncte adiționale P_1, \dots, P_t ale lui $f(x)$;

4.6. calculează mesajul de verificare $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i, k_0), i = 1, \dots, t$;

4.7. transmite, $i = 1, \dots, t$:

$$KGC \rightarrow U_i : (P_1, \dots, P_t, V_i)$$

Calculul Cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

5.1. verifică dacă $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i, k_0)$;

Dacă egalitatea nu se satisface, abandonează;

5.2. calculează cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și

$$(pw_{ix}, pw_{iy} + k_i).$$

Figura 5.12: Prima versiune îmbunătățită a protocolului Yuan et al.'s Protocol [10]

Pas 1. U_a inițializează 4 sesiuni (s_j) , $j = 1, \dots, 4$, cerând stabilirea unei chei comune cu U_i ;

Pas 2. U_a este un utilizator autorizat pentru toate cele 4 sesiuni, așa încât determină:

$$f(x)_{(s_j)} = a_{(s_j)}x^2 + b_{(s_j)}x + c_{(s_j)}, j = 1, \dots, 4$$

Pas 3. Cum $(pw_{ix}, pw_{iy} + k_{i(s_j)})$ sunt puncte valide ale lui $f(x)_{(s_j)}$, U_a obține:

$$pw_{iy} + k_{i(s_j)} = a_{(s_j)}pw_{ix}^2 + b_{(s_j)}pw_{ix} + c_{(s_j)}, j = 1, \dots, 4$$

Pas 4. U_a interceptează $K_{i(s_j)}$, știe că $k_{i(s_j)} = K_{i(s_j)} - pw_{ix}$ și determină:

$$pw_{iy} = a_{(s_j)}pw_{ix}^2 + (b_{(s_j)} + 1)pw_{ix} + c_{(s_j)} - K_{i(s_j)}, j = 1, \dots, 4$$

Pas 5. U_a elimină pw_{iy} din primele 2 ecuații ($j = 1, 2$), respectiv din ultimele 2 ($j = 3, 4$) și obține:

$$\begin{aligned} A_{(s_1s_2)}pw_{ix}^2 + B_{(s_1s_2)}pw_{ix} + C_{(s_1s_2)} &= 0 \\ A_{(s_3s_4)}pw_{ix}^2 + B_{(s_3s_4)}pw_{ix} + C_{(s_3s_4)} &= 0 \end{aligned}$$

unde:

$$\begin{aligned} A_{(s_1s_2)} &= a_{(s_1)} - a_{(s_2)} & A_{(s_3s_4)} &= a_{(s_3)} - a_{(s_4)} \\ B_{(s_1s_2)} &= b_{(s_1)} - b_{(s_2)} & B_{(s_3s_4)} &= b_{(s_3)} - b_{(s_4)} \\ C_{(s_1s_2)} &= c_{(s_1)} - c_{(s_2)} - (K_{i(s_1)} - K_{i(s_2)}) & C_{(s_3s_4)} &= c_{(s_3)} - c_{(s_4)} - (K_{i(s_3)} - K_{i(s_4)}) \end{aligned}$$

Pas 6. U_a determină parola de lungă durată a lui U_i ca:

$$\begin{aligned} pw_{ix} &= (A_{(s_1s_2)}C_{(s_3s_4)} - A_{(s_3s_4)}C_{(s_1s_2)})(A_{(s_3s_4)}B_{(s_1s_2)} - A_{(s_1s_2)}B_{(s_3s_4)})^{-1} \\ pw_{iy} &= f(pw_{ix})_{(s_j)} - K_{i(s_j)} + pw_{ix}, \text{ for any } j = 1, \dots, 4 \end{aligned}$$

Figura 5.13: Atac din interior asupra primei versiuni îmbunătățite a protocolului Yuan et al. [15]

Inițializare. KGC alege 2 numere prime mari p și q și calculează $n = pq$;

Înregistrare. Fiecare utilizator $U_i, i = 1, \dots, m$, stabilește o parolă de lungă durată $pw_i = pw_{ix} || pw_{iy}$ cu KGC ;

Runda 1. Utilizatorul U_1 :

1.1. transmite o cerere de generare a cheii:

$$U_1 \rightarrow KGC : \{U_1, \dots, U_t\}$$

Runda 2. KGC :

2.1. alege $k_0 \leftarrow^R \mathbb{Z}_n$;

2.2. transmite:

$$KGC \rightarrow^* : (\{U_1, \dots, U_t\}, k_0)$$

Runda 3. Fiecare utilizator $U_i, i = 1, \dots, t$:

3.1. alege $k_i \leftarrow^R \mathbb{Z}_n$;

3.2. calculează $K_i = pw_{ix} + k_i$ și $M_i = h_1(U_1, \dots, U_t, k_i, k_0)$;

3.3. transmite:

$$U_i \rightarrow KGC : (U_i, \{U_1, \dots, U_t\}, K_i, M_i)$$

Runda 4. KGC :

4.1. calculează $k_i = K_i - pw_{ix}, i=1, \dots, t$;

4.2. verifică dacă $M_i = h_1(U_1, \dots, U_t, k_i, k_0), i=1, \dots, t$;

Dacă egalitatea nu se satisface, abandonează;

4.3. alege 2 numere aleatoare x_{ta} și y_{ta} de lungime egală cu pw_{ix} și pw_{iy} ;

4.4. generează polinomul $f(x)$ de grad t care trece prin $t + 1$ puncte $(x_{ta}, y_{ta}), (pw_{1x}, h_3(U_1, \dots, U_t, pw_{1y}, k_1, k_0)), \dots, (pw_{tx}, h_3(U_1, \dots, U_t, pw_{ty}, k_t, k_0))$;

4.5. calculează t puncte adiționale P_1, \dots, P_t ale lui $f(x)$;

4.6. calculează mesajul de verificare $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i, k_0), i = 1, \dots, t$;

4.7. transmite, $i = 1, \dots, t$:

$$KGC \rightarrow U_i : (P_1, \dots, P_t, V_i)$$

Calculul cheii. Fiecare utilizator $U_i, i = 1, \dots, t$:

5.1. verifică dacă $V_i = h_2(U_1, \dots, U_t, P_1, \dots, P_t, k_i, k_0)$;

Dacă egalitatea nu se satisface, abandonează;

5.2. calculează cheia de grup $k = f(0)$ prin interpolarea punctelor P_1, \dots, P_t și

$$(pw_{ix}, h_3(U_1, \dots, U_t, pw_{iy}, k_i, k_0)).$$

Figura 5.14: A doua versiune îmbunătățită a protocolului Yuan et al. [10]

Capitolul 6

Protocele GKE formal sigure bazate pe SSS

6.1 Un nou protocol

Propunem un nou protocol GKA bazat pe scheme de m -partajare perfecte, descris în detaliu în Fig.6.1 [20].

6.2 Demonstrații de securitate

Protocolul propus este demonstrat sigur în cadrul modelului formal de securitate GBG. Detalii cu privire la semnificația noțiunilor și demonstrația teoremelor se regăsesc în varianta completă a lucrării.

Teorema 6.1. (*Securitatea AKE*¹) [20]. Dacă schema de semnătură electronică Σ este UF-CMA², funcția cu trapă secretă \mathcal{F} este sigură, funcția hash H este un oracol aleator și schema de m -partajare \mathcal{SS} este perfectă, atunci protocolul propus este AKE sigur și

$$\text{Adv}_{\mathcal{A}}^{\text{AKE}} \leq 2m^2 \text{Adv}_{\mathcal{A}, \Sigma}^{\text{UF-CMA}} + \frac{(q_s + q_r)^2}{2^{\mathcal{K}-1}} + \frac{(m+1)q_s^2}{2^{\mathcal{K}-1}} + 2\text{Adv}_{\mathcal{A}, \mathcal{F}}.$$

Teorema 6.2. (*MA Security*³) [20]. Dacă schema de semnătură electronică Σ este UF-CMA și funcția hash H este un oracol aleator, atunci protocolul propus este MA sigur și

$$\text{Adv}_{\mathcal{A}}^{\text{MA}} \leq m^2 \text{Adv}_{\mathcal{A}, \Sigma}^{\text{UF-CMA}} + \frac{(q_s + q_r)^2}{2^{\mathcal{K}}} + \frac{(m+1)q_s^2}{2^{\mathcal{K}}}.$$

¹Authenticated Key Exchange.

²UnForgeable under Chosen Message Attack.

³Mutual Authentication.

Inițializare. Fie $\mathcal{F} = (\text{Gen}, \text{F}, \text{F}^{-1})$ o funcție cu trapă secretă și SS o schemă de m -partajare a secretelor.

Înregistrarea. Fie $\Sigma.\text{Sig}_{U_i}$ algoritmul de semnătură utilizat de U_i și $\Sigma.\text{Verify}_{U_i}$ algoritmul corespunzător de verificare, $i = 1, \dots, m$.

Runda 1. Utilizatorul U_1 :

- 1.1. execută $\mathcal{F}.\text{Gen}$ pentru a obține o pereche cheie publică - cheie privată (pk, sk) ;
- 1.2. alege $r_1 \leftarrow^R \{0, 1\}^{\mathcal{K}}$;
- 1.3. transmite:

$$U_1 \rightarrow^*: (\mathcal{U}, pk, \text{F}(pk, r_1), \sigma = \Sigma.\text{Sign}_{U_1}(\mathcal{U}||pk||\text{F}(pk, r_1)));$$

Runda 2. Fiecare utilizator U_i , $i = 2, \dots, m$:

- 2.1. verifică dacă $\Sigma.\text{Verify}_{U_1}(\mathcal{U}||pk||\text{F}(pk, r_1), \sigma) = 1$.
Dacă egalitatea nu se satisface, abandonează;
- 2.2. alege $r_i \leftarrow^R \{0, 1\}^{\mathcal{K}}$;
- 2.3. transmite:

$$U_i \rightarrow^*: (\text{F}(pk, r_i), \sigma_i = \Sigma.\text{Sign}_{U_i}(\mathcal{U}||pk||\text{F}(pk, r_i)));$$

Runda 3. Utilizatorul U_1 :

- 3.1. verifică dacă $\Sigma.\text{Verify}_{U_i}(\mathcal{U}||pk||\text{F}(pk, r_i), \sigma_i) = 1$, $i = 2, \dots, m$.
Dacă cel puțin o egalitate nu se satisface, abandonează;
- 3.2. calculează $\text{sid}_{U_1} = \text{F}(pk, r_1)||\dots||\text{F}(pk, r_m)$, $r_i = \text{F}^{-1}(sk, \text{F}(pk, r_i))$, cheia de sesiune $k = \text{H}(\text{sid}_{U_1}||r_1||r_2||\dots||r_m)$ și r'_i a.î. $SS.\text{Share}(k, 2) = \{r_i, r'_i\}$, $i = 2, \dots, m$;
- 3.3. transmite:

$$U_1 \rightarrow^*: (r'_2, \dots, r'_m, \sigma' = \Sigma.\text{Sign}_{U_1}(\mathcal{U}||pk||r'_2||\dots||r'_m||\text{sid}_{U_1}));$$

Calculul cheii. Fiecare utilizator U_i , $i = 2, \dots, m$:

- 4.1. verifică dacă $\Sigma.\text{Verify}_{U_j}(\mathcal{U}||pk||\text{F}(pk, r_j), \sigma_j) = 1$, $j = 2, \dots, m$, $j \neq i$.
Dacă cel puțin o egalitate nu se satisface, abandonează;
- 4.2. calculează $\text{sid}_{U_i} = \text{F}(pk, r_1)||\dots||\text{F}(pk, r_m)$ și $k = SS.\text{Rec}(r_i, r'_i)$;
- 4.3. verifică dacă $\Sigma.\text{Verify}_{U_1}(\mathcal{U}||pk||r'_2||\dots||r'_m||\text{sid}_{U_i}, \sigma') = 1$.
Dacă se satisface egalitatea, acceptă cheia k ; altfel, abandonează;

Confirmarea cheii. Fiecare utilizator U_i , $i = 2, \dots, m$:

- 5.1. calculează r_j a.î. $SS.\text{Share}(k, 2) = \{r_j, r'_j\}$, $j = 2, \dots, m$, $j \neq i$;
- 5.2. verifică dacă $\text{F}(pk, r_j)$ este egal cu valoare transmisă în pasul 2.3.
Dacă cel puțin o egalitate nu se satisface, abandonează;

Figura 6.1: Protocolul de agreare a cheilor de grup Olimid [20]

Tabelul 6.1: Analiza complexității protocolului propus [20]

	Stocare	Calculabilitate	Transmisie
U_1	$l_{sk} + l_{sk_1} + \mathcal{K}$	$c_{\mathcal{F}.Gen} + c_{\mathcal{F}} + (m-1)c_{\mathcal{F}-1} + c_{\mathcal{H}} + 2c_{\Sigma.Sign} + (m-1)c_{\Sigma.Verify} + (m-1)c_{SS.Share}$	$l_{\mathcal{U}} + l_{pk} + ml_{\mathcal{F}} + (m+1)l_{\Sigma.Sign} + (m-1)\mathcal{K}$
U_i ($i \neq 1$)	$l_{sk_i} + \mathcal{K}$	$c_{\mathcal{F}} + c_{\Sigma.Sign} + mc_{\Sigma.Verify} + c_{SS.Rec} + (m-2)c_{SS.Share}$	

Tabelul 6.2: Comparația protocolului propus cu protocoale existente [20]

	Nr. de runde	Tip de transmisiune	Grup static/dinamic	Generare sid	Model de securitate
Protocolul propus	3	broadcast	static	la rulare	GBG/ROM
Bresson-Catalano [2]	3	unicast, broadcast	static	anterior	BCP/ROM
Cao et al. [3]	3	broadcast	static	anterior	UC/ROM

Teorema 6.3. (*Contributivitate*) [20]. Dacă funcția cu trapă secretă \mathcal{F} este sigură și funcția hash \mathcal{H} este un oracol aleator, atunci protocolul propus satisface proprietatea de contributivitate și

$$\text{Adv}_{\mathcal{A}}^{\text{Con}} \leq \frac{(m+1)q_s^2}{2\mathcal{K}} + \frac{q_r}{2\mathcal{K}}.$$

6.3 Analiză

Tabelul 6.1 analizează complexitatea protocolului propus din trei perspective distincte: capacitate necesară de stocare, cost computațional și cost total de transmisie. Am notat cu l_x lungimea (în biți) a lui x și cu c_y costul de execuție a lui y .

Tabelul 6.2 compară protocolul propus cu două protocoale GKA existente, care beneficiază de asemenea de o demonstrație guroasă de securitate, însă în modele distincte.

Concluzii și direcții viitoare de cercetare

Teza de față se axează asupra protocoalelor de stabilire a cheilor de grup bazate pe partajarea secretelor și schemelor de partajare utilizate pentru construcția acestora. Menționăm pe scurt rezultatele personale și indicăm posibile direcții viitoare de cercetare. Precizăm că toate capitolele, cu excepția celor introductive (Partea I, Capitolul 1 și Partea II, Capitolul 4), conțin rezultate proprii. Împărțim rezultatele obținute în două direcții de studiu:

- **Criptanaliză (Capitolele 3 și 5).**

Introducem un atac SETUP asupra schemelor de partajare a secretelor care utilizează destule valori aleatoare, cu rolul de a oferi unui atacator un avantaj copleșitor în determinarea secretului [14]. Deși implementări malițioase au fost implementate pentru multiple primitive criptografice (protocoale de generare a cheilor, semnături electronice, sisteme de vot electronic, etc.), suntem primii care le considerăm în cazul partajării de secrete. Tehnica propusă poate fi aplicată unor scheme clasice, precum Shamir, care este utilizată pentru construcția a numeroase protocoale de stabilire a cheilor de grup. În consecință, analiza securității acestor protocoale cu privire la atacul de tip SETUP asupra schemelor pe care se bazează reprezintă un subiect imediat de cercetare.

Motivați de multitudinea protocoalelor de stabilire a cheilor de grup bazate pe scheme de partajare a secretelor definite în ultimii ani și care nu beneficiază de o demonstrație riguroasă de securitate, propunem o analiză a acestora [10], [15], [16], [17], [18], [19]. Cum nu există argumente riguroase care să le ateste securitatea, vulnerabilitățile tind să apară în mod natural. Introducem șase astfel de atacuri asupra a patru protocoale foarte recente (sau a variantelor lor îmbunătățite) (2012-2013) și propunem modalități de remediere a acestora. Versiunilor îmbunătățite pentru a rezista la aceste atacuri le lipsește de asemenea o demonstrație de securitate, ceea ce înseamnă că pot fi de asemenea ușor vulnerabile. Cercetări viitoare pot extinde succesiunile de atacuri și contramăsuri aplicate protocoalelor analizate sau pot dezvălui atacuri similare asupra altor protocoale.

- **Construcții (Capitolele 2 și 6).**

Definim o schemă de partajare vizuală care permite împărțirea unei imagini alb-negru secrete în componente color [11], [12]. Propunerea nu oferă nici o informație despre secret pentru cel mult doi participanți și permite reconstrucția perfectă când toți

participanții colaborează. O construcție a unui protocol de stabilire a cheilor de grup bazat pe scheme vizuale de partajare poate reprezenta o abordare interesantă pentru cercetări viitoare. Atât cât cunoaștem, nu există o astfel de construcție în literatură. Suntem conștienți de limitările practice ale unui astfel de propuneri, dar putem indica o utilizare imediată: partajarea unei chei vizuale secrete comune care poate fi ulterior utilizată în cadrul criptării vizuale.

Prezentăm noțiunea de *scheme de m-partajare* și introducem conceptul de *scheme de m-partajare perfecte* ca o generalizare a schemelor de partajare perfecte [20]. Cele foarte puține protocoale de stabilire a cheilor de grup bazate pe scheme de partajare care dețin o analiză riguroasă a securității reprezintă o motivație: folosim schemele de *m-partajare* (datorită avantajelor pe care le introduc) ca primitivă pentru construcția unui protocol GKA [20]. Protocolul este demonstrat sigur în cadrul modelului GBG și menține același număr de runde de comunicație ca și protocoalele existente. În limita cunoștințelor noastre, nici un alt protocol bazat pe scheme de partajare nu este demonstrat sigur într-un model de securitate similar. Considerăm ca direcții viitoare de cercetare îmbunătățirea protocolului pentru a rezista la atacuri EKL⁴ (o proprietate care nu este modelată în GBG), a admite grupuri dinamice, anonimitate, robustețe sau eficiență crescută. Definirea unor noi protocoale sigure de stabilire a cheilor de grup care utilizează partajarea secretelor reprezintă o sarcină destul de dificilă.

Alte posibile direcții de cercetare includ:

- *definirea protocoalelor GKE flexibile bazate pe scheme de partajare a secretelor*. Protocoalele flexibile GKE au fost introduse în 2010 de către Abdalla et al. [1]: în timp ce un grup de utilizatori stabilește o cheie secretă comună, fiecare membru obține informație suplimentară pe care o poate utiliza ulterior pentru a dobândi, la cerere și fără a iniția o nouă sesiune, chei independente partajate de submulțimi ale mulțimii inițiale de participanți.
- *definirea protocoalelor GKE asimetrice (ASGKE) bazate pe scheme de partajare a secretelor*. Protocoalele asimetrice GKE au fost introduse în 2009 de către Wu et al. [25]: un grup de utilizatori partajează o cheie publică comună, dar fiecare membru deține o cheie privată distinctă (care îi permite de exemplu să decripteze mesajele criptate folosind cheia publică comună sau să genereze o semnătură verificabilă cu cheia publică comună). Aceste sisteme prezintă proprietăți remarcabile: monitorizare, delegare a atribuțiilor, păstrarea cheilor de rezervă, generarea semnăturilor de grup. Menționez că dețin un rezultat personal în acest domeniu, însă construcția nu se bazează pe scheme de partajare a secretelor, drept urmare nu face parte direct din aria de interes a tezei de doctorat [13].
- *analizarea și îmbunătățirea protocoalelor GKE existente bazate pe primitive și concepte moderne*. Lucrarea de față se rezumă la protocoale GKE bazate pe scheme de partajare clasice. O direcție viitoare de studiu o constituie domeniul protocoalelor care utilizează scheme de partajare bazate pe latici, perechi biliniare sau chiar cuantice.

⁴Ephemeral Key Leakage.

Bibliografie

- [1] Michel Abdalla, Céline Chevalier, Mark Manulis, and David Pointcheval. Flexible group key exchange with on-demand computation of subgroup keys. In *Proceedings of the Third international conference on Cryptology in Africa, AFRICACRYPT'10*, pages 351–368, Berlin, Heidelberg, 2010. Springer-Verlag. pages 48
- [2] Emmanuel Bresson and Dario Catalano. Constant round authenticated group key agreement via distributed computation. In *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography, PKC'2004*, pages 115–129. Springer, 2004. pages 45
- [3] Chunjie Cao, Chao Yang, Jianfeng Ma, and Sangjae Moon. Constructing UC secure and constant-round group key exchange protocols via secret sharing. *EURASIP J. Wirel. Commun. Netw.*, 2008:4:1–4:9, January 2008. pages 45
- [4] Lien Harn and Changlu Lin. Authenticated group key transfer protocol based on secret sharing. *IEEE Trans. Comput.*, 59(6):842–846, June 2010. pages 29, 36
- [5] Chingfang Hsu, Bing Zeng, Qi Cheng, and Guohua Cui. A novel group key transfer protocol. Cryptology ePrint Archive, Report 2012/043, 2012. pages 29, 32
- [6] Ehud D. Karnin, Jonathan W. Greene, and Martin E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29:35–41, 1983. pages 16
- [7] Mijin Kim, Namje Park, and Dongho Won. Cryptanalysis of an authenticated group key transfer protocol based on secret sharing. In *Grid and Pervasive Computing*, pages 761–766. Springer-Verlag, 2013. pages 29
- [8] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., 1996. pages 13, 14, 27, 28
- [9] Junghyun Nam, Moonseong Kim, Juryon Paik, Woongryul Jeon, Byunghee Lee, and Dongho Won. Cryptanalysis of a group key transfer protocol based on secret sharing. In *Proceedings of the Third international conference on Future Generation Information Technology, FGIT'11*, pages 309–315, Berlin, Heidelberg, 2011. Springer-Verlag. pages 29, 36
- [10] Ruxandra F. Olimid. A chain of attacks and countermeasures applied to a group key transfer protocol (abstract). In *(Pre)Proceedings of Western European Workshop on Research in Cryptology, WEWoRC'13*. pages 29, 33, 36, 39, 40, 42, 47

- [11] Ruxandra F. Olimid. About a visual secret sharing scheme. In *Proceedings of the 3rd International Conference on Security for Information Technology and Communications, SECITC'10*, pages 13–17, 2010. pages 16, 17, 19, 47
- [12] Ruxandra F. Olimid. Python implementation of visual secret sharing schemes. *Journal of Information Systems & Operations Management*, 5(2.1):538–550, December 2011. pages 17, 47
- [13] Ruxandra F. Olimid. A new public key cryptosystem with key escrow capabilities. In *Proceedings of the 7th South East European Doctoral Student Conference, DSC 2012*, pages 610–624, 2012. pages 48
- [14] Ruxandra F. Olimid. SETUP in secret sharing schemes using random number generators. 2012. Under review. pages 22, 23, 24, 47
- [15] Ruxandra F. Olimid. Cryptanalysis of a password-based group key exchange protocol using secret sharing. *Appl. Math. Inf. Sci.*, 7(4):1585–1590, July 2013. pages 29, 33, 36, 41, 47
- [16] Ruxandra F. Olimid. On the (in)security of group key transfer protocols. In *Proceedings of the Romanian Academy, series A*, volume 14, Special Issue RCD'13, pages 378–387, 2013. pages 47
- [17] Ruxandra F. Olimid. On the (non)improvement of an authenticated group key transfer protocol based on secret sharing. 2013. Under review. pages 29, 31, 47
- [18] Ruxandra F. Olimid. On the security of an authenticated group key transfer protocol based on secret sharing. In *Proceedings of the 2013 international conference on Information and Communication Technology, ICT-EurAsia'13*, pages 399–408, Berlin, Heidelberg, 2013. Springer-Verlag. pages 29, 33, 36, 37, 47
- [19] Ruxandra F. Olimid. On the vulnerability of a group key transfer protocol based on secret sharing. 2013. Under review. pages 29, 31, 33, 34, 47
- [20] Ruxandra F. Olimid. Provable secure constant-round group key agreement protocol based on secret sharing. In *Proceedings of International Joint Conference SOCO'13 - CISIS'13 - ICEUTE'13*, AISC 239, pages 489–498. Springer, 2013. pages 15, 43, 44, 45, 48
- [21] Joseph Pieprzyk and Chih-Hung Li. Multiparty key agreement protocols. *IEE Proceedings-Computers and Digital Techniques*, 147(4):229–236, 2000. pages 28, 33
- [22] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979. pages 29, 33
- [23] Yi Sun, Qiaoyan Wen, Hongxiang Sun, Wenmin Li, Zhengping Jin, and Hua Zhang. An authenticated group key transfer protocol based on secret sharing. *Procedia Engineering*, 29(0):403 – 408, 2012. 2012 International Workshop on Information and Electronics Engineering. pages 16, 29, 33, 35

- [24] Web-site. Python programming language, Official website. Last accessed: July 2013. <http://www.python.org/>. pages 18
- [25] Qianhong Wu, Yi Mu, Willy Susilo, Bo Qin, and Josep Domingo-Ferrer. Asymmetric group key agreement. In *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '09, pages 153–170, Berlin, Heidelberg, 2009. Springer-Verlag. pages 48
- [26] Adam Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 89–103, London, UK, 1996. Springer-Verlag. pages 21
- [27] Adam Young and Moti Yung. Kleptography: using cryptography against cryptography. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 62–74, Berlin, Heidelberg, 1997. Springer-Verlag. pages 21
- [28] Wei Yuan, Liang Hu, Hongtu Li, and Jianfeng Chu. An efficient password-based group key exchange protocol using secret sharing. *Appl. Math. Inf. Sci.*, 7(1):145–150, January 2013. pages 29, 33, 38
- [29] Wei Yuan, Liang Hu, Hongtu Li, and Jianfeng Chu. Security and improvement of an authenticated group key transfer protocol based on secret sharing. *Appl. Math. Inf. Sci.*, 7(5):1943–1949, September 2013. pages 29, 30