HU 0002239 20 B1

(19) HU

MAGYAR KÖZTÁRSASÁG
Magyar Szabadalmi Hivatal

(11) Lajstromszám: 223 920 (13) B1

SZABADALMI LEÍRÁS

(21) A bejelentés ügyszáma: P 01 04054
(22) A bejelentés napja: 1999. 09. 22.

(40) A közzététel napja: 2002. 03. 28.
(45) A megadás meghirdetésének dátuma a Szabadalmi Közlöny és Védjegyértesítőben: 2005. 03. 29.

(51) Int. Cl.[7]: H 04 L 9/08

(86) A nemzetközi (PCT) bejelentési szám: PCT/EP 99/07051
(87) A nemzetközi közzétételi szám: WO 0022775

(30) Elsőbbségi adatok:
198 47 941.7 1998. 10. 09. DE

(72) Feltaláló:
Schwenk, Jörg, Dieburg (DE)

(73) Jogosult:
Deutsche Telekom AG, Bonn (DE)

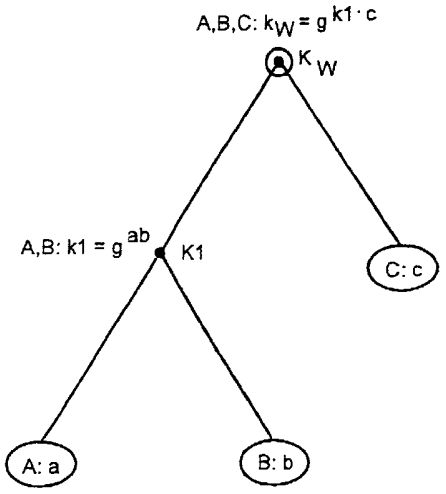
(74) Képviselő:
Kovári György, ADVOPATENT Szabadalmi és Védjegy Iroda, Budapest

(54) **Eljárás közös kriptográfiai kulcs létrehozására n felhasználó számára**

(57) Kivonat

A találmány szerinti eljárással közös kriptográfiai kulcsot hoznak létre n felhasználó számára DH-eljárás alkalmazásával. Az n számú felhasználó mindegyikéhez egy n levelű és $[\log_2 n]$ mélységű bináris fa egy-egy levelét rendelik hozzá. Az eljárás során

- mindegyik felhasználó (A, B, C) számára véletlen számként (a, b, c) egy titkot generálnak, és ezt a fának ahhoz a leveléhez rendelik hozzá, amelyhez az illető felhasználó (A, B, C) van hozzárendelve, továbbá
- a fa gyökere (K_w) felé haladva egymás után a fa összes csomópontja (K1) számára közös kulcsként (k1) titkokat generálnak, ennek során a levelekből kiindulva az adott fastruktúrának megfelelően a fa teljes hierarchiájában mindig két-két már ismert titokból a DH-eljárással egy-egy új titkot hoznak létre, ezeket egy-egy közös csomóponthoz rendelik hozzá, és végül az utolsó csomóponthoz – a fa gyökeréhez (K_w) – az összes felhasználó közös kulcsát (k_w) tartalmazó titkot rendelik hozzá. Az eljárás segítségével a csoportkulcs létrehozása után is könnyen törölhetők felhasználók a kulcsjegyzékből, vagy abba bevihetők.



1. ábra

A leírás terjedelme 8 oldal (ezen belül 3 lap ábra)

A találmány tárgya eljárás közös kriptográfiai kulcs létrehozására n felhasználó számára olyan üzenetek titkosságának biztosításához, amelyeket nem biztonságos kommunikációs csatornákon át kizárólag az említett n számú felhasználóhoz kívánunk átvinni.

Két vagy több személy közötti kommunikáció bizalmasságának és sértetlenségének védelméhez titkosítási és hitelesítési mechanizmusokat alkalmaznak. Ezek végrehajtásához azonban szükség van arra, hogy az összes felhasználó rendelkezzen egy közös információval. Ez a közös információ a kriptográfiai kulcs.

Nem biztonságos kommunikációs csatornákon át folytatott átvitelhez közös kulcs létrehozására Diffie és Hellman dolgozott ki egy elméletet (DH-eljárás; W. Diffie and M. Hellman, New Direetions in Cryptography, IEEE Transactions on Information Theory, IT–22 (6): 644–654, november 1976).

A Diffie–Hellman-féle kulccsere alapja az, hogy nagy p prímszám esetén gyakorlatilag lehetetlen a diszkrét logaritmus kiszámítása. Ezt használja ki Alice és Bob az alábbi példában, ahol mindketten titokban egy-egy x, illetve y számot választanak, amelyek kisebbek p-nél (és relatív prímek p-1-gyel). Ezután (egyszerre vagy egymás után) elküldik egymásnak egy nyilvános α szám x-edik (ill. y-odik) hatványát. A beérkezett hatványokból x-szel, illetve y-nal végzett újabb hatványozással egy közös $K:=\alpha^{xy}$ kulcsot számítanak ki. A csak az α^x és α^y értékeket látó támadó ezekből nem tudja kiszámítani a K kulcsot. (Az egyetlen jelenleg ismert módszer szerint először például α^x α modulo p alapú logaritmusát kellene kiszámítani, majd α^y-t erre a hatványra emelni.)

Példa a Diffie–Hellman-kulcscserére:

Alice	Bob
Titkosan x-et választ	Titkosan y-t választ
α^x →	
	α^y ←

Kulcsképzés: $K:=(\alpha^y)^x=\alpha^{xy}$ $K:=(\alpha^x)^y=a^{xy}$

A példában leírt DH-kulcscserével kapcsolatban az a probléma, hogy Alice nem tudja, hogy tényleg Bobbal kommunikál, és nem egy csalóval. Az IPSec-ben ezt a problémát nyilvánoskulcs-tanúsítvány alkalmazásával oldják meg, amelyben egy megbízható tanúsító hatóság a felhasználó azonosságát egy nyilvános kulcshoz rendeli hozzá. Ezzel azonosítható a partner.

A DH-kulcscsere más matematikai struktúrákkal is megoldható, például $GF(2^n)$ véges testekkel vagy elliptikus görbékkel. Ezekkel az alternatívákkal javítható a teljesítmény. Ez az eljárás azonban csak arra alkalmas, hogy két felhasználó megállapodjon a kulcsban.

Különböző kísérletek történtek arra, hogy a DH-eljárást kiterjesszék három vagy több felhasználóra (DH-csoportok). (A technika állásáról áttekintést ad M. Steiner, G. Tsudik, M. Waidner, Diffie–Hellman Key Distribution Extended to Group Communication. Proc. 3rd ACM Conference on Computer and Communications Security, March 1996, New Delhi, India.)

A DH-eljárás kiterjesztését három A, B és C felhasználóra az alábbi táblázat szemlélteti. (Mindig modulo p aritmetikával számolva):

	A→B	B→C	C→A
1. kör	g^a	g^b	g^c
2. kör	g^{ca}	g^{ab}	g^{bc}

A fenti két kör után a három felhasználó mindegyike képes kiszámítani a g^{abc} mod p titkos kulcsot.

Ezeknek a kiterjesztéseknek mindegyikénél fellép az alábbi három probléma legalább egyike:

- A felhasználóknak bizonyos módon, például – mint a fenti példában – körben kell elhelyezkedni.
- A felhasználóknak a központtal szemben nincs befolyásuk a kulcs kiválasztására.
- A körök száma a felhasználók számától függ.

A DE 195 38 385.0 számú iratból egy további eljárás ismeretes kulcs közös generálására. Ennél az eljárásnál a központnak ismernie kell legalább a felhasználók titkos kulcsait.

Ismeretes egy további eljárás is közös kriptográfiai kulcs létrehozására (David A. McGrew, Alan T. Sherman: Key establishment in large dynamic groups using one-way function trees, IEEE Transaction On Software Engineering, 1998. 05. 20., 1–13. old.). Ez az eljárás egy fastruktúrán alapul. Egy csoportmenedzser egy bináris fát kezel, ahol mindegyik x csomóponthoz két kriptográfiai kulcsot rendel hozzá, mégpedig egy kx csomóponti kulcsot és egy rejtett $k'x \approx g(k_x)$ csomóponti kulcsot. A rejtett csomóponti kulcsot egy egyirányú függvénnyel lehet kiszámítani a csomóponti kulcsból. Mindegyik felhasználó ismeri a nem rejtett csomóponti kulcsokat a saját csomópontjától a gyökérig vezető útvonalon, valamint az ezzel szomszédos csomópontok rejtett csomóponti kulcsait, egyébként nem ismer semmilyen más rejtett vagy nem rejtett kulcsot. Ennek az eljárásnak a végrehajtása nyilvánvalóan azon alapszik, hogy a csoportmenedzser ismeri a fa leveleihez tartozó összes kulcsot.

Egy további ismert megoldásnál (Burmeister, Desmedt: A secure and efficient conference key distribution system, Proc. EUROCRYPT'94, Springer LNCS, Berlin 1994) a kulcs generálásához két kör vagy forduló szükséges, amelyek közül a második körben a központi állomás n számú, p=kb. 1000 bit hosszúságú üzenetet küld n számú felhasználónak.

Ismeretes még egy (n, t) küszöbérték-eljárásnak (küszöbsémának) nevezett kriptográfiai eljárás (titokmegosztási séma). Ezzel az eljárással egy k kulcs úgy osztható szét t részre (árnyékra), hogy a k kulcs a t számú árnyék közül tetszőlegesen kiválasztott n számú árnyékból rekonstruálható (Beutelspacher, Schwenk, Wolfenstetter: Moderne Verfahren der Kryptographie, 2. Auflage, Vieweg Verlag, Wiesbaden 1998).

Célunk a találmánnyal egy közös csoportkulcs előállítása egy központi állomás és egy n számú felhasználóból álló csoport között. Az eljárásnak olyannak kell

lennie, hogy felhasználók a csoportkulcs létrehozása után is különösebb nehézség nélkül törölhetők legyenek a kulcsjegyzékből, vagy abba bevihetők legyenek.

A találmány szerinti eljárással közös kriptográfiai kulcsot hozunk létre n felhasználó számára DH-eljárás alkalmazásával. Az n számú felhasználó mindegyikéhez egy pontosan n levelű és közelítően [$\log_2 n$] mélységű bináris fa egy-egy levelét rendeljük hozzá. Az eljárás során

- mindegyik felhasználó számára egy titkot generálunk, és ezt a fának ahhoz a leveléhez rendeljük hozzá, amelyhez az illető felhasználó van hozzárendelve, továbbá
- a fa gyökere felé haladva egymás után a fa összes csomópontja számára titkokat generálunk, ennek során a levelekből kiindulva az adott fastruktúrának megfelelően a fa teljes hierarchiájában mindig két-két már ismert titokból a DH-eljárással egy-egy új titkot hozunk létre, ezeket egy-egy közös csomóponthoz rendeljük hozzá, és végül az utolsó csomóponthoz – a fa gyökeréhez – az összes felhasználó közös kulcsát tartalmazó titkot rendeljük hozzá.

Amikor egy új felhasználót illesztünk be egy közös titokkal rendelkező, már meglevő fastruktúrába, az n+1 felhasználó közös kulcsának előállításához a bináris fa egy megfelelő pontján lévő levél helyett két új levelet illesztünk be, és ezzel egy n+1 levelű, [$\log_2(n+1)$] mélységű új fát állítunk elő; a meglévő levélhez hozzárendelt felhasználót az egyik új levélhez, az új felhasználót a másik új levélhez rendeljük hozzá, és az eddigi levél helyén a két új levél számára egy közös csomópontot hozunk létre; továbbá az új levelekből kiindulva a fa gyökeréig csak azokban a csomópontokban hozunk létre új titkokat, amelyek a fastruktúrában az új levelektől a fa gyökeréig vezető úton helyezkednek el.

Amikor egy felhasználót kizárunk egy közös titokkal rendelkező, már létező fastruktúrából, mind a kizárandó felhasználó levelét, mind az ugyanahhoz a közös csomóponthoz tartozó másik felhasználó levelét eltávolítjuk; továbbá a közös csomópontot a megmaradó másik felhasználó levelévé alakítjuk, és a fa leveleiből kiindulva a gyökérig csak azokban a csomópontokban hozunk létre új titkokat, amelyek a fastruktúrában az új levéltől a fa gyökeréig vezető úton helyezkednek el.

A találmányt a továbbiakban kiviteli példák és rajzok alapján részletesen ismertetjük. A rajzokon az

1. ábra: a találmány szerinti kulcsgenerálás elvét szemlélteti, három felhasználót tartalmazó fastruktúra esetén, a
2. ábra: a kulcsgenerálás négy felhasználót tartalmazó fastruktúrában, a
3. ábra: a kulcsgenerálás öt felhasználót tartalmazó fastruktúrában, a
4. ábra: a 2. ábra szerinti, már létező fastruktúra bővítése egy további felhasználóval, és az
5. ábra: egy felhasználó eltávolítása a 2. ábra szerinti, már létező fastruktúrából.

Az 1. ábra a találmány szerinti eljárás elvét szemlélteti három A, B, C felhasználót tartalmazó fastruktúrában. Egy közös kulcs létrehozásához az A, B, C felhasználók a következőképpen járnak el:

- Az A és B felhasználók egy DH-eljárást hajtanak végre véletlenszerűen generált a és b számokkal. Előállítanak egy közös $k1=g^{ab}$ mod p kulcsot, amelyet egy közös K1 csomóponthoz rendelnek hozzá.
- Az egyik oldalon az A és B felhasználók, és a másik oldalon a C felhasználó egy második DH-eljárást hajtanak végre, az A és B felhasználók közös k1 kulcsa és a C felhasználó véletlenszerűen generált c száma alapján. Az eredmény egy közös $k=g^{k1\cdot c}$ mod p kulcs, amelyet a fa K_w gyökeréhez rendelnek hozzá.

A 2. ábra négy A, B, C, D felhasználó esetén mutatja be a közös kulcs előállítását. Ennek során az A, B, C és D felhasználók a következőképpen járnak el:

- Az A és B felhasználók egy DH-eljárást hajtanak végre véletlenszerűen generált a és b számokkal. Ennek eredményeként egy közös $k1=g^{ab}$ mod p kulcsot kapnak.
- A C és D felhasználók egy DH-eljárást hajtanak végre véletlenszerűen generált c és d számokkal. Az eredmény egy közös $k2=g^{cd}$ mod p kulcs.
- Az egyik oldalon az A és B felhasználók, és a másik oldalon a C és D felhasználók egy második DH-eljárást hajtanak végre, az A és B felhasználók közös k1 kulcsa és a C és D felhasználók közös k2 kulcsa alapján. Az eredmény egy közös $k_w=g^{k1\cdot k2}$ mod p kulcs, amelyet a fa K_w gyökeréhez rendelnek hozzá.

A 3. ábra öt A, B, C, D, E felhasználó esetén mutatja be a kulcsgenerálást. Egy közös kulcs létrehozásához az A, B, C, D és E felhasználók a következőképpen járnak el:

- Az A és B felhasználók egy DH-eljárást hajtanak végre véletlenszerűen generált a és b számokkal. Kapnak egy közös $k1=g^{ab}$ mod p kulcsot.
- A C és D felhasználók egy DH-eljárást hajtanak végre véletlenszerűen generált c és d számokkal. Kapnak egy közös $k2=g^{cd}$ mod p kulcsot.
- Az egyik oldalon az A és B felhasználók, és a másik oldalon a C és D felhasználók egy második DH-eljárást hajtanak végre, az A és B felhasználók közös k1 kulcsa és a C és D felhasználók közös k2 kulcsa alapján. Az eredmény az A, B, C és D felhasználók közös $k_3=g^{k1\cdot k2}$ mod p kulcsa.
- Az egyik oldalon az A, B, C és D felhasználók, és a másik oldalon az E felhasználó egy harmadik DH-eljárást hajtanak végre, az A, B, C és D felhasználók közös k3 kulcsa és az E felhasználó véletlen e száma alapján. Az eredmény a közös $k_w=g^{k3\cdot e}$ mod p kulcs, amelyet a fa K_w gyökeréhez rendelnek hozzá.

A találmány szerinti eljárás lehetővé teszi új felhasználók bevonását, vagy egyes felhasználók kizárását anélkül, hogy minden felhasználóval újra végre kellene hajtani az egész eljárást.

Egy új felhasználó bevonását a 4. ábra alapján írjuk le. A 2. ábra szerinti fastruktúrából indulunk ki, amelyhez a B felhasználónak megfelelő levélnél csat-

lakozik egy új felhasználó. Amikor egy közös titokkal rendelkező, már létező fastruktúrához kell hozzáadni egy újabb felhasználót, az n+1 számú felhasználó számára egy új közös kulcs létrehozásához a bináris fa egy alkalmas pontján (jelen esetben a B felhasználó levelénél) két új levelet kell beilleszteni. Az új fa tehát n+1 levelű, és $[\log_2(n+1)]$ mélységű lesz. Az eddigi B felhasználót B1 felhasználóként az egyik új levélhez rendeljük hozzá. Az új B2 felhasználót a másik, még szabad új levélhez rendeljük hozzá. A korábbi B felhasználó levele helyén a B1 és B2 felhasználókhoz tartozó K1 csomópont lesz. A B1 és B2 felhasználóknak megfelelő új levelekből kiindulva a fa K_w gyökere felé csak azokban a csomópontokban keletkeznek új titkok, amelyek a B1 és B2 felhasználók új leveleitől a fa K_w gyökeréhez vezető úton helyezkednek el. Jelen esetben tehát a K1 és K2 csomópontokról és a K_w gyökérről van szó.

Ha a felhasználók száma kettő valamelyik hatványának felel meg, a fa mélysége (szintjeinek száma) az új felhasználó hozzáadása miatt eggyel nő (ld. a fenti példát). Ha a felhasználók száma nem kettő hatványa, a felosztandó levél ügyes kiválasztásával elkerülhető a mélység növekedése, amint a következő példa mutatja:

Amikor például három felhasználó mellé egy negyediket kívánunk bevonni a rendszerbe, a következőképpen járunk el (az 1. ábra szerinti helyzetből kiindulva):

- A C felhasználó a bekapcsolódó új D felhasználóval egy DH-eljárást folytat le véletlen c' és d számokkal (c'-nek lehetőleg különböznie kell a korábban választott c-től). Az eredmény $k2'=g^{c'd} \bmod p$.
- Az egyik oldalon az A és B felhasználók, és a másik oldalon a C és D felhasználók egy DH-eljárást hajtanak végre a k1 és k2 kulcsokkal. Az eredmény $k\text{-}g^{k1 \cdot k2'} \bmod p$.

Egy ilyen konfigurációnál az A és B felhasználóknak nem kell új kulcscserét végrehajtani. Általánosságban csak azokat a titkokat kell újra egyeztetni, amelyek a fán az új felhasználónak megfelelő levéltől a K_w gyökérig vezető úton helyezkednek el.

Egy felhasználó kizárását vagy törlését az 5. ábra szerinti, négy felhasználót tartalmazó fastruktúra alapján ismertetjük. Itt a kiindulási helyzet a 2. ábra szerinti fastruktúrának felel meg, amelyből a B felhasználót kell eltávolítani.

Amikor egy B felhasználót kizárunk vagy törlünk egy közös titokkal rendelkező, már meglevő fastruktúrából, mind a törlendő B felhasználónak megfelelő levelet, mind az ugyanahhoz a közös K1 csomóponthoz tartozó A felhasználónak megfelelő levelet eltávolítjuk, amint az 5. ábrán látható. A közös K1 csomópontból egy új A' levél lesz, amely a fastruktúrában maradó A felhasználónak felel meg. A fa leveleitől a K_w gyökérig haladva csak azokban a csomópontokban keletkeznek új titkok, amelyek a fastruktúrában közvetlenül az új A' levél és a K_w gyökér között helyezkednek el. Példánkban egyetlen ilyen csomópont van, ez a K_w gyökér. Egy ilyen konfigurációnál a C és D felhasználóknak nem kell új kulcscserét végrehajtani. Általában tehát itt is csak azokban a titkokban kell újra megállapodni, amelyek a fastruktúrában az új felhasználónak megfelelő levéltől a K_w gyökérig vezető úton helyezkednek el.

Az eljárás előnyösen többféle módon is továbbfejleszthető:

Az $x \rightarrow g^x$ diszkrét exponenciális függvény képzéséhez például más csoportok is felhasználhatók.

Egy felhasználó beillesztésénél vagy eltávolításánál például meg lehet egyezni abban, hogy a DH-eljárás szükséges újabb végrehajtásához nem a régi titkokat, hanem egy (esetleg randomizált) egyirányú függvény eredményét használjuk fel.

SZABADALMI IGÉNYPONTOK

1. Eljárás közös kriptográfiai kulcs létrehozására n felhasználó számára DH-eljárás alkalmazásával, amelynél az n számú felhasználó mindegyikéhez egy pontosan n levelű és közelítően $[\log_2 n]$ mélységű bináris fa egy-egy levelét rendeljük hozzá,

azzal jellemezve, hogy

- mindegyik felhasználó számára egy titkot generálunk, és ezt a fának ahhoz a leveléhez rendeljük hozzá, amelyhez az illető felhasználó van hozzárendelve, továbbá
- a fa gyökere felé haladva egymás után a fa összes csomópontja számára titkokat generálunk, ennek során a levelekből kiindulva az adott fastruktúrának megfelelően a fa teljes hierarchiájában mindig két-két már ismert titokból a DH-eljárással egy-egy új titkot hozunk létre, ezeket egy-egy közös csomóponthoz rendeljük hozzá, és végül az utolsó csomóponthoz – a fa gyökeréhez – az összes felhasználó közös kulcsát tartalmazó titkot rendeljük hozzá.

2. Az 1. igénypont szerinti eljárás,

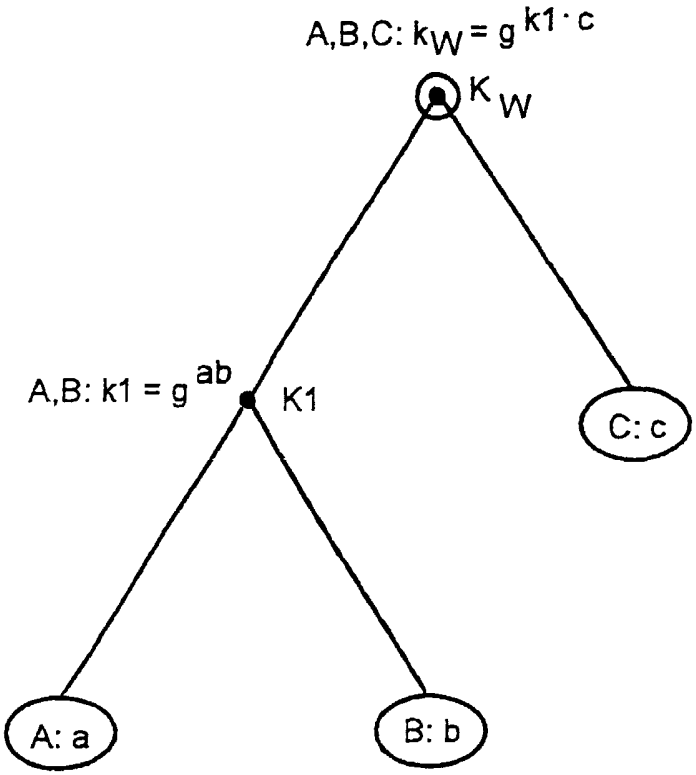
azzal jellemezve, hogy

- amikor egy új felhasználót illesztünk be egy közös titokkal rendelkező, már meglevő fastruktúrába, az n+1 felhasználó közös kulcsának előállításához a bináris fa egy megfelelő pontján lévő levél helyett két új levelet illesztünk be, és ezzel egy n+1 levelű, $[\log_2(n+1)]$ mélységű új fát állítunk elő,
- a meglévő levélhez hozzárendelt felhasználót az egyik új levélhez, az új felhasználót a másik új levélhez rendeljük hozzá, és az eddigi levél helyén a két új levél számára egy közös csomópontot hozunk létre, továbbá

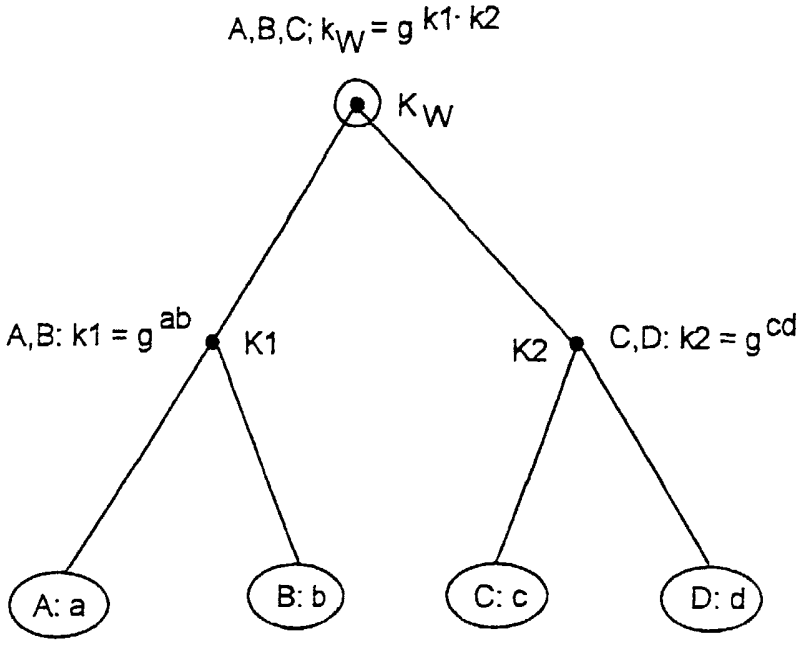
– az új levelekből kiindulva a fa gyökeréig csak azokban a csomópontokban hozunk létre új titkokat, amelyek a fastruktúrában az új levelektől a fa gyökeréig vezető úton helyezkednek el.

3. Az 1. igénypont szerinti eljárás,
azzal jellemezve, hogy

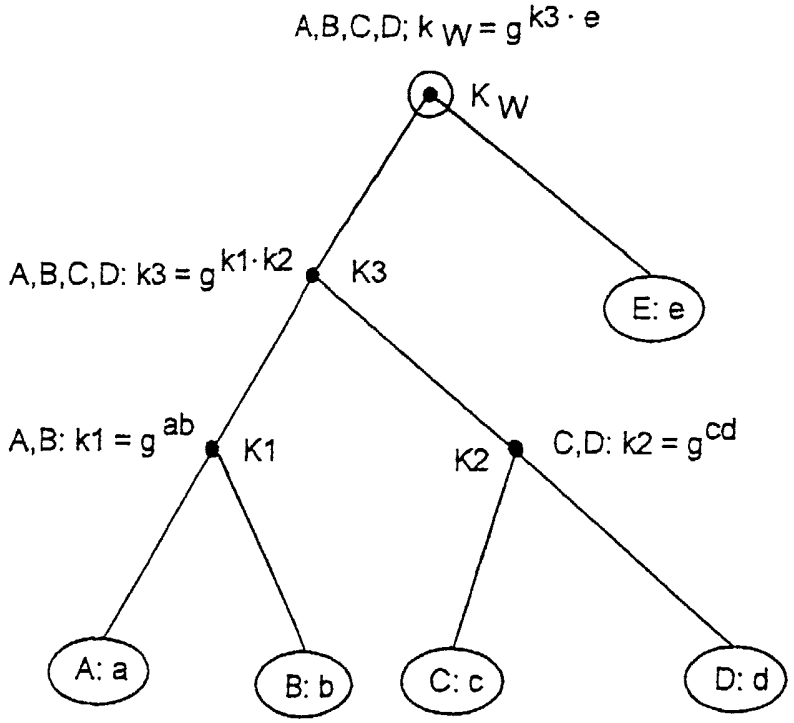
– amikor egy felhasználót kizárunk egy közös titokkal rendelkező, már létező fastruktúrából, mind a kizárandó felhasználó levelét, mind az ugyanahhoz a közös csomóponthoz tartozó másik felhasználó levelét eltávolítjuk, továbbá
– a közös csomópontot a megmaradó másik felhasználó levelévé alakítjuk, és a fa leveleiből kiindulva a gyökérig csak azokban a csomópontokban hozunk létre új titkokat, amelyek a fastruktúrában az új levéltől a fa gyökeréig vezető úton helyezkednek el.



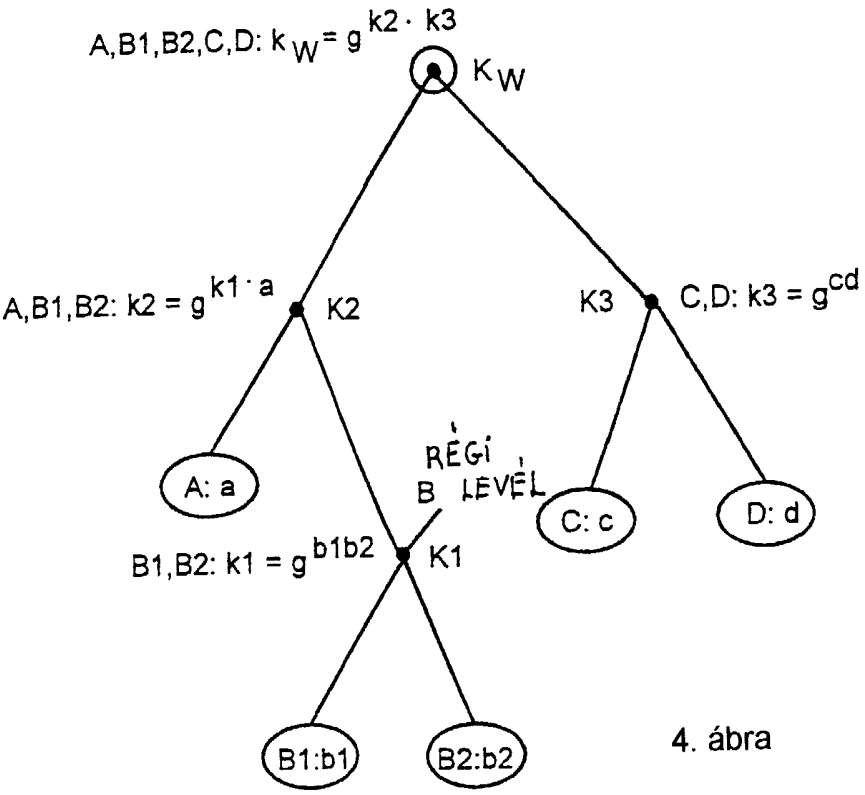
1. ábra


A,B,C; $k_W = g^{k1 \cdot k2}$
K_W
A,B: $k1 = g^{ab}$
K1
K2
C,D: $k2 = g^{cd}$
A: a
B: b
C: c
D: d

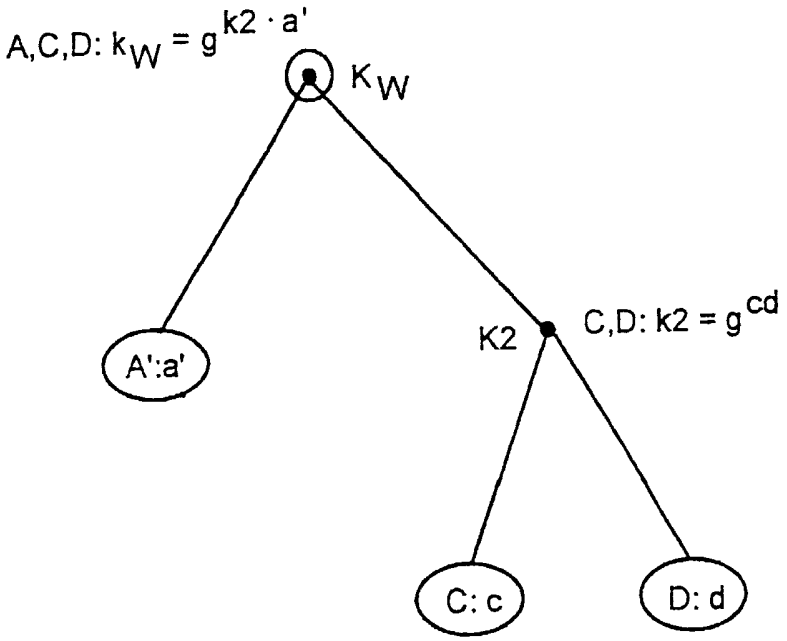
2. ábra


A,B,C,D; $k_W = g^{k3 \cdot e}$
K_W
A,B,C,D: $k3 = g^{k1 \cdot k2}$
K3
E: e
A,B: $k1 = g^{ab}$
K1
K2
C,D: $k2 = g^{cd}$
A: a
B: b
C: c
D: d

3. ábra



4. ábra



5. ábra

Kiadja a Magyar Szabadalmi Hivatal, Budapest
A kiadásért felel: Törőcsik Zsuzsanna főosztályvezető-helyettes
Windor Bt., Budapest