~~73.817/KOT~~

Kivonat

Eljárás kliens és forrás szerver közötti biztonságos kommunikációhoz átkódoló proxyval

Az eljárás annak lehetővé tételére szolgál, hogy proxy (15) részt vegyen kliens (10') és első forrás szerver (12a') közötti biztonságos kommunikációban. A találmány értelmében:

(a) első biztonságos összeköttetést építenek fel ~~(20, 22)~~ a kliens és a proxy között;

(b) az első biztonságos összeköttetés ellenőrzése után második biztonságos összeköttetést építenek fel ~~(24, 26)~~ a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön az első forrás szerverhez;

(c) a kliens és az első forrás szerver megbeszélnek ~~(30)~~ egy összeköttetés mester titkot;

(d) a kliens eljuttatja ~~(32)~~ az összeköttetés mester titkot a proxyhoz az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt vegyen a biztonságos kommunikációban;

(e) válaszként kliens oldali kérésre az első forrás szerverhez, harmadik biztonságos összeköttetést építenek fel a kliens és a proxy között, amely harmadik biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön a második forrás szerverhez (12b');

(f) a kliens és a második forrás szerver megbeszélnek új összeköttetés mester titkot; és

(g) a kliens eljuttatja az új összeköttetés mester titkot a proxynak az (a) lépésben generált első biztonságos összeköttetés használatával.

/2. ábra/

P0201706

S. B. G. & K.
Szabadalmi Ügyvivői Iroda
H-1062 Budapest, Andrássy út 113.
Telefon: 461-1000, Fax: 461-1099

73.817/KOT

A

KÖZZÉTÉTELI PÉLDÁNY

Eljárás kliens és forrás szerver közötti biztonságos kommunikációhoz átkódoló proxyval

A jelen találmány általánosan hálózati biztonsági protokollokkal kapcsolatos, szűkebb értelemben pedig eljárással egy kliens és egy vagy több forrás szerver közötti biztonságos összeköttetés bizalmasságának kiterjesztésére egy közvetítőre (intermediary) (például egy átkódoló proxyra).

A hálózati biztonsági protokollok, mint a Netscape Biztonságos Kapcsolati Réteg (SSL - Secure Socket Layer) protokollja és az IETF (Internet Engineering Task Force) Szállítási Réteg Biztonság (TLS - Transport Layer Security) protokoll, biztosítják a bizalmasságot és az adatintegritást kommunikáló alkalmazások között. Ezeket a protokollokat széles körben használják például az Interneten keresztül történő elektronikus kereskedelem biztonságossá tételére.

M. Norifusa („Internet security: difficulties and solutions", International Journal of Medical Informatics, 1998. március, 67-74. oldalak) a 71-73. oldalakon bemutatja a SOCKS 5 protokoll leírását.

A közelmúltban a számítógép ipar egyik célja az volt, hogy számítógépes feldolgozási és kommunikációs képességekkel lásson el olyan eszközöket, amelyek különböznek azoktól az

eszközöktől, amiket normális esetben hagyományos számítógépeknek tekintenénk. Az ilyen eszközök meglehetősen változatosak, és magukban foglalják például a személyi digitális asszisztenseket (PDA - personal digital assistant), üzleti szervezőket (például az **IBM**® WorkPad® és a 3Com® PalmPilot®), intelligens telefonokat, mobiltelefonokat, más kézi készülékeket és hasonlókat. Kényelmi okokból ezekre az eszközökre mint osztályra néha „átjárható számítási" kliensekként („pervasive computing" clients) hivatkozunk, mivel ezek olyan eszközök, amelyeket azzal a céllal terveztek, hogy egy számítógépes hálózatban lévő szerverekhez legyenek kapcsolva és számítási célokra legyenek használhatóak az elhelyezkedésüktől függetlenül.

Az átjárható számítási kliensek viszont tipikusan nem támogatják egy HTML (Hypertext Markup Language, hipertext jelölőnyelv) Windows-alapú kliens teljes funkció-készletét. Ennek eredményeképpen tipikusan szükségünk van átkódoló szolgáltatásokra az információ lefordításához az egyik forrás jelölőnyelvről (például HTML-ről) egy másikra (például HDML-re) (HDML - handheld device markup language, kézi készülék jelölőnyelv), az átjárható kliensen elérendő információ lefordítására. Az átkódoló szolgáltatások biztosítása valamely biztonságos hálózati kapcsolaton keresztül viszont problematikus. Különösen azért, mert alapvető konfliktus áll fenn a biztonsági és az átkódoló szolgáltatások között, mivel a hagyományos biztonsági protokollokat, például az SSL-t és a TLS-t is pontosan azért tervezték, hogy megakadályozzanak egy harmadik felet ab-

ban, hogy beleavatkozzon a kliens és a szerver közötti kommunikációba.

Még szűkebb értelemben a jelen találmány lehetővé teszi egy proxy számára, hogy átkódoló szolgáltatást nyújtson, miközben egy átjárható számítási kliens egy vagy több forrás szerverrel kommunikál biztonságos összeköttetésen keresztül.

A jelen találmány lehetővé teszi továbbá egy proxy számára ideiglenes tárolás vagy más adminisztratív szolgáltatások végrehajtását egy olyan kliens nevében, amely egy vagy több szerverrel kommunikál hálózati biztonsági protokoll használatával.

A jelen találmány lehetővé teszi ezen kívül egy proxy számára, hogy titkosítást/visszafejtést hajtson végre egy olyan kliens nevében, amely egy vagy több forrás szerverrel kommunikál egy hálózati szerver protokoll használatával.

Egy előnyös kiviteli példában egy proxy részt vesz egy kliens és egy első szerver közötti biztonságos kommunikációban. Az eljárás azzal kezdődik, hogy felépítünk egy első biztonságos összeköttetést a kliens és a proxy között. Az első biztonságos összeköttetés ellenőrzése után az eljárás azzal folytatódik, hogy felépítünk egy második biztonságos összeköttetést a kliens és a proxy között. A második biztonságos összeköttetésben a kliens kéri a proxyt, hogy csatornaként (conduit) működjön az első szerverhez. Ezután a kliens és az első szerver megbeszél egy első összeköttetés mester titkot. Az első biztonságos összeköttetés használatával a kliens ezután biztosítja a proxyhoz ezt az első összeköttetés mester

titkot, lehetővé téve a proxy számára, hogy részt vegyen a kliens és az első szerver közötti biztonságos kommunikációkban. Miután fogadta az első összeköttetés mester titkot, a proxy kriptográfiai (titkosítási) információt generál, amely lehetővé teszi számára, hogy valamely adott szolgáltatást (például átkódolást, monitorozást, titkosítást/visszafejtést, ideiglenes tárolást vagy hasonlót) biztosítson a kliens nevében anélkül, hogy a szerver tudna a részvételéről. Az első biztonságos összeköttetést fenntartjuk a kliens és a proxy között az ilyen kommunikációk közben.

A jelen találmány egy jellemzője szerint, ha a proxynak adatra van szüksége egy második szervertől egy adott kliens oldali kérés feldolgozásához, a fent említett protokollt ismételjük meg. Vagyis a proxy kibocsát egy kérést a klienshez egy különálló kapcsolat felépítésére a második szerverrel, ismét a proxyn keresztül való alagutazás (tunneling) által. Amint azt fent leírtuk, ez a protokoll lehetővé teszi a kliens számára, hogy egy második összeköttetés mester titkot alakítson ki a második szerverrel, és ezt a titkot megosztja a proxyval az előzőekben leírt módon. A proxy ezután folytatja a szolgáltatási működését (például az átkódolást) ezen második titok használatával bizalmas adat megszerzéséhez a második szervertől.

Ily módon, ha már egyszer az alap alagút protokoll felépült a kliens és egy adott forrás szerver között, a kliens szükség szerint megismétli a protokollt, hogy lehetővé tegye a proxy számára akár „n" további forrás szervertől származó biz-

tonságos adat megszerzését, miközben az egy adott kliens oldali kérést szolgáltat az adott forrás szerverhez.

A jelen találmány eljárást biztosít annak lehetővé tételére egy proxy számára, hogy részt vegyen egy kliens és egy első forrás szerver közötti biztonságos kommunikációban, amely eljárás magában foglalja a következő lépéseket: felépítünk egy első biztonságos összeköttetést a kliens és a proxy között; az első biztonságos összeköttetés ellenőrzése után felépítünk egy második biztonságos összeköttetést a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön az első forrás szerverhez; a kliens és az első forrás szerver megbeszél egy összeköttetés mester titkot; a kliens eljuttatja az összeköttetés mester titkot a proxynak az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt vegyen a biztonságos kommunikációban; válaszként egy kliens oldali kérésre az első forrás szerverhez, felépítünk egy harmadik biztonságos összeköttetést a kliens és a proxy között, amely harmadik biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön a második forrás szerverhez; a kliens és a második forrás szerver megbeszél egy új összeköttetés mester titkot; végül a kliens továbbítja az új összeköttetés mester titkot a proxynak az első biztonságos összeköttetés használatával.

A jelen találmány eljárást biztosít továbbá annak lehetővé tételére egy proxy számára, hogy részt vegyen egy kliens és egy szerver közötti biztonságos kommunikációban, amely eljárást egy kliensben használjuk 1-„n" szerverek halmazával való

kommunikáláshoz, és az eljárás magában foglalja a következő lépéseket: az 1-„n" szerverek halmazának mindegyike esetén: (1) a kliens egy első biztonságos összeköttetést kér a proxyhoz; (2) a proxytól fogadott valamely tanúsítvány érvényességének hitelesítése után a kliens egy második biztonságos összeköttetést kér a proxyhoz, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön a szerverhez; (3) a kliens és a szerver megbeszél egy vonatkozó összeköttetés mester titkot a csatornán keresztül; (4) a megbeszélés befejezése után a kliens eljuttatja a vonatkozó összeköttetés mester titkokat a proxynak az első biztonságos összeköttetés használatával; valamint a proxy a vonatkozó összeköttetés mester titkokat használja olyan adott kriptográfiai információ generálására, amely hasznos a biztonságos kommunikációban való részvételhez.

A jelen találmány eljárást biztosít ezen kívül annak lehetővé tételére egy proxy számára, hogy részt vegyen egy biztonságos kommunikációban, az eljárást egy kliensben használjuk, és magában foglalja a következő lépéseket: egy kérést továbbítunk egy klienstől a proxyhoz egy első biztonságos összeköttetés felépítésére; egy kérést továbbítunk a klienstől a proxyhoz egy második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy forrás szerverhez; egy összeköttetés mester titkot továbbítunk a klienstől a proxyhoz az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt ve-

gyen a biztonságos kommunikációban; válaszként a proxytól származó valamely kérés fogadására a kliensnél, egy kérést továbbítunk a klienstől a proxyhoz egy harmadik biztonságos összeköttetés felépítésére a kliens és a proxy között, amely harmadik biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy másik forrás szerverhez; és átviszünk egy új összeköttetés mester titkot a klienstől a proxyhoz.

A jelen találmány eljárást biztosít továbbá annak lehetővé tételére egy proxy számára, hogy részt vegyen egy biztonságos kommunikációban, az eljárást egy proxyban használjuk, és magában foglalja a következő lépéseket: a proxynál egy klienstől származó kérést fogadunk egy első biztonságos összeköttetés felépítésére a kliens és a proxy között; a proxynál a klienstől származó kérést fogadunk egy második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy forrás szerverhez; a proxynál a klienstől az első biztonságos összeköttetés használatával továbbított összeköttetés mester titkot fogadunk; egy adott kérés továbbítása után a proxytól a klienshez, a proxynál a klienstől származó kérést fogadunk egy második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy másik forrás szerverhez; és a proxynál a klienstől továbbított új összeköttetés mester titkot fogadunk.

A jelen találmány eljárást biztosít ezen kívül annak lehetővé tételére egy proxy számára, hogy részt vegyen egy ösz-

szeköttetésben egy kliens és egy első forrás szerver között, amely eljárást egy proxyban használjuk, és magában foglalja a következő lépéseket: a proxynál egy klienstől származó kérést fogadunk egy első biztonságos összeköttetés felépítésére a kliens és a proxy között; a proxyn keresztül egy biztonsági kézfogási eljárást vezetünk a kliens és az első forrás szerver között egy első összeköttetés kulcs létrehozására; a kliens továbbítja az első összeköttetés kulcsot a proxyhoz, ami által a proxy részt vehet a kliens és az első forrás szerver közötti kommunikációban az összeköttetés közben; és az összeköttetés folytatásában egy biztonsági kézfogási eljárást vezetünk a kliens és egy második forrás szerver között egy második összeköttetés kulcs létrehozására; és a kliens továbbítja a második összeköttetés kulcsot a proxyhoz, ami által a proxy adatot szerezhet meg a második forrás szervertől, a klienstől az első forrás szerverhez való kérés kiszolgálásakor való használatához.

A jelen találmány biztosít továbbá egy kriptográfiai rendszert, amely magában foglal: egy klienst; szerverek egy halmazát; egy proxyt; egy hálózati protokoll szolgáltatást, amely lehetővé teszi a kliens és minden egyes szerver számára, hogy egy biztonságos kapcsolaton keresztül kommunikáljanak; egy számítógépes programot (i) a kliens vezérlésére, hogy egy első biztonságos kapcsolatot kérjen a proxyhoz, (ii) válaszként a proxytól származó valamely tanúsítvány érvényességének hitelesítésére, a kliens vezérlésére, hogy egy második biztonságos kapcsolatot kérjen a proxyhoz, amely második biztonságos

kapcsolat kéri a proxyt, hogy csatornaként működjön egy adott szerverhez, (iii) a kliens vezérlésére, hogy megbeszélést folytasson az adott szerverrel a csatornán keresztül egy összeköttetés mester titok megszerzéséhez, és (iv) a megbeszélés sikeres befejezése után a kliens vezérlésére, hogy eljuttassa az összeköttetés mester titkot a proxyhoz az első biztonságos kapcsolat használatával; és egy számítógépes programot (i) a proxy vezérlésére, hogy az összeköttetés mester titkot használja adott kriptográfiai információ generálására, (ii) a proxy vezérlésére, hogy kérje, hogy a kliens szelektív módon építsen fel egy különálló biztonságos kapcsolatot egy másik szerverrel, és (iii) a proxy átkapcsolására egy aktív működési állapotba, amely közben részt vehet a kliens és az adott szerver közötti kommunikációkban.

A jelen találmány biztosít ezen kívül számítógépes program terméket egy számítógép által olvasható közegen egy klienst, szerverek egy halmazát és egy proxyt tartalmazó kriptográfiai rendszerben való használathoz, amely magában foglal: egy első rutint (i) a kliens vezérlésére, hogy egy első biztonságos kapcsolatot kérjen a proxyhoz, (ii) válaszként a proxytól származó valamely tanúsítvány érvényességének hitelesítésére, a kliens vezérlésére, hogy egy második biztonságos kapcsolatot kérjen a proxyhoz, amely második biztonságos kapcsolat kéri a proxyt, hogy csatornaként működjön egy adott szerverhez, (iii) a kliens vezérlésére, hogy megbeszélést folytasson az adott szerverrel a csatornán keresztül egy összeköttetés mester titok megszerzéséhez, és (iv) a megbeszélés

sikeres befejezése után a kliens vezérlésére, hogy eljuttassa az összeköttetés mester titkot a proxyhoz az első biztonságos kapcsolat használatával; és egy második rutint (i) a proxy vezérlésére, hogy az összeköttetés mester titkot használja adott kriptográfiai információ generálására, (ii) a proxy vezérlésére, hogy kérje, hogy a kliens szelektív módon építsen fel egy különálló biztonságos kapcsolatot egy másik szerverrel, és (iii) a proxy átkapcsolására egy aktív működési állapotba, amely közben részt vehet a kliens és az adott szerver közötti kommunikációkban.

A jelen találmány biztosít továbbá számítógép által olvasható program kóddal rendelkező számítógépes program terméket egy használható hordozón egy kliensben való használathoz annak lehetővé tételére egy proxy számára, hogy részt vegyen egy biztonságos kommunikációban, amely magában foglal: eszközt egy kérés továbbítására a klienstől a proxyhoz egy első biztonságos összeköttetés felépítésére; eszközt egy kérés továbbítására a klienstől a proxyhoz egy második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy forrás szerverhez; eszközt egy összeköttetés mester titok továbbítására a klienstől a proxyhoz az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt vegyen a biztonságos kommunikációban; eszközt a kliens vezérlésére, hogy megszerezzen egy új összeköttetés mester titkot válaszként a proxytól származó valamely adott kérés fogadására a kliensnél a biztonságos kommu-

nikáció közben; és eszközt az új összeköttetés mester titok továbbítására a klienstől a proxyhoz.

A jelen találmány biztosít ezen kívül számítógép által olvasható program kóddal rendelkező számítógépes program terméket egy használható hordozón egy proxyban való használathoz annak lehetővé tételére a proxy számára, hogy részt vegyen egy biztonságos kommunikációban, amely magában foglal: eszközt egy klienstől származó kérés fogadására a proxynál egy első biztonságos összeköttetés felépítésére a kliens és a proxy között; eszközt egy klienstől származó kérés fogadására a proxynál egy második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön egy forrás szerverhez; eszközt a klienstől továbbított valamely összeköttetés mester titok fogadására a proxynál az első biztonságos összeköttetés használatával; eszközt egy adott kérés továbbítására a proxytól a klienshez válaszként egy adott előforduláshoz a biztonságos kommunikáció közben; és eszközt a klienstől továbbított valamely új összeköttetés mester titok fogadására a proxynál.

A következőkben leírjuk a találmányt, csak példák gyanánt, a rajzokra való hivatkozással, amelyeken:

Az 1. ábra egy hálózati biztonsági protokollt használó ismert kliens-szerver hálózati környezet egy egyszerűsített diagramja;

A 2. ábra egy kliens-szerver hálózati környezet egy egyszerűsített diagramja, ahol egy harmadik fél közvetítő eszköz vagy proxy részt vesz egy biztonságos összeköttetésben;

A 3A. és 3B. ábra együtt az alap alagutazási eljárás egy részletes folyamatlistát tartalmazó ábrája;

A 4. ábra a jelen találmány egy egyszerűsített blokk diagramja, ahol, miután a kliens kezdetben biztonságot helyez ki a proxyhoz, a proxy kéri a klienst, hogy építsen fel egy vagy több további biztonságos kapcsolatot a proxyn keresztül való alagutazás által „n" további forrás szerverhez; és

Az 5. ábra egy átjárható számítási kliens-szerver architektúra egy blokk diagramja, amelyben a jelen találmány megvalósítható.

Az 1. ábra egy eddig használatos, hagyományos kliens-szerver hálózati architektúrát ábrázol. Ebben az ábrázolásban a 10 kliens kommunikál a 12 szerverrel egy 14 hálózaton keresztül, amely lehet az Internet, egy intranet, egy nagy kiterjedésű hálózat, egy lokális kiterjedésű hálózat vagy hasonló. A 10 kliens és a 12 szerver egy hálózati biztonsági protokoll használatával kommunikál, például a Netscape Biztonságos Kapcsolati Réteg (SSL) protokolljával vagy az IETF Szállítási Réteg Biztonság (TLS) protokolljával. Általánosítva, egy kliens lehet bármely alkalmazás entitás, amely egy TLS vagy SSL kapcsolatot kezdeményez egy szerverhez. Egy szerver lehet bármely alkalmazás entitás vagy program, amely kapcsolatokat fogad el kérések kiszolgálására válaszok visszaküldésével. Bármely adott program képes lehet arra, hogy egyszerre legyen

mind kliens, mind pedig szerver. Az elsődleges működési különbség a szerver és a kliens között az, hogy a szerver általában hitelesített, míg a kliens csak opcionálisan hitelesített. A szerverre, amelyen egy adott erőforrás elhelyezkedik vagy azon létrehozandó, a továbbiakban forrás szerverként hivatkozunk.

A 10 kliens és a 12 szerver egy biztonságos összeköttetésben vesz részt. Egy SSL vagy TLS összeköttetés egy megfeleltetés egy kliens és egy szerver között, amelyet egy kézfogási protokollal hozunk létre. Az összeköttetések kriptográfiai biztonsági paraméterek egy halmazát definiálják, amelyeket megoszthatunk több összeköttetés között. Ezeket arra használjuk, hogy elkerüljük az új biztonsági paraméterek költséges megbeszélését minden egyes összeköttetés esetén. Az SSL-ben vagy TLS-ben egy összeköttetés azonosító egy szerver által generált érték, amely egy meghatározott összeköttetést azonosít. Egy SSL vagy TLS összeköttetés felépítéséhez a kliens és a szerver kézfogást hajt végre, ami egy olyan kezdeti megbeszélés, amely kialakítja az entitások közötti valamely tranzakció paramétereit. Ha már egy összeköttetést létrehoztunk, a kliens és a szerver közötti kommunikációk egy kapcsolaton keresztül történnek, ami egy olyan szállítás (az OSI rétegező modell definícióban), amely egy alkalmas típusú szolgáltatást biztosít. Az SSL és TLS esetén az ilyen kapcsolatok egyedek közötti (peer-to-peer) kapcsolatok. A kapcsolat tranziens (átmeneti), és mindegyik kapcsolatot egyetlen összeköttetésnek feleltetjük meg. A kapcsolaton keresztüli kommunikációkat tipikusan nyil-

vános kulcsú titkosítás által tesszük biztonságossá, amely kétkulcsos titkosítókat alkalmazó kriptográfiai technikák egy osztálya. Valamely nyilvános kulccsal titkosított üzeneteket csak egy társított privát kulccsal fejthetünk vissza. Megfordítva, a privát kulccsal aláírt üzeneteket a nyilvános kulccsal ellenőrizhetjük.

Ha már egyszer az összeköttetést felépítettük, a kliens rendelkezik egy tanúsítvánnyal, amelyet a forrás szerver bocsátott ki abból a célból, hogy az hitelesítse a klienst a forrás szerver felé. A kliens is kéri a forrás szervert, hogy az bemutasson egy tanúsítványt, ami alapján a forrás szervert érvényesként hitelesítheti. A hitelesítés egy entitás azon képessége, hogy meghatározza egy másik entitás kilétét. Tipikusan, az X.509 protokoll (másként az ISO Hitelesítési keret (Authentication framework)) részeként, a tanúsítványokat egy megbízható tanúsítvány hatóság adja ki, és ezek egy erős kötést biztosítanak egy fél identitása (vagy valamely más attribútuma) és a nyilvános kulcsa között.

A fent leírt funkcionalitás ismeretes a tárgykörben. A funkcionalitást például olyan protokollokban valósítjuk meg, amelyek megfelelnek az IETF TLS 1.0 verziónak és az SSL 2.0/3.0 verziónak. Ezek a protokollok, bár nagyon hasonlóak, két rétegből állnak: a rekord protokollból és a kézfogási protokollból. Amint azt látni fogjuk, a jelen találmány felhasználja ezen típusú biztonsági protokoll kiterjesztésének egy eljárását egy összeköttetés bizalmasságának kiterjesztésére egy harmadik fél közvetítő eszközre vagy proxyra. Előnyösen a

találmányt egy kliens és egy proxy közötti kézfogási eljárással együtt valósítjuk meg, amely, amint azt látni fogjuk, egy biztonságos összeköttetés legfelső rétegén helyezkedik el. Ez a kiterjesztés nem változtatja meg a biztonságos kapcsolat alap tulajdonságait a rekord protokoll rétegnél. Bár a technikát TLS és SSL összefüggésében írjuk le, ez nem tekintendő a jelen találmány korlátozásának.

A 2. ábrára hivatkozva, az alap eljárás lehetővé teszi a 10' kliens számára, amely SSL-t vagy TLS-t használ biztonsági protokollként egy vagy több 12'a-n forrás szerverrel való kommunikáláshoz, hogy megengedje egy 15 proxy számára az összeköttetésben való részvételt az összeköttetés biztonsági attribútumainak megváltoztatása nélkül. Amint azt fent megjegyeztük, ez az eljárás független a titkosítás erősségétől vagy a 10' kliens és egy adott 12' forrás szerver által egymás hitelesítésére végrehajtott lépésektől. A jelen találmány a TLS/SSL ugyanazon előnyeivel rendelkezik, ami abban áll, hogy kiterjeszti a protokollt, mégis megengedi magasabb szintű protokolloknak a legfelső rétegre való helyezését. Az ilyen magasabb szintű protokollok magukban foglalnak például olyan alkalmazás protokollokat (például HTTP-t, TELNET-et, FTP-t és SMTP-t), amelyek normális esetben közvetlenül a szállítási (például a TCP/IP) réteg fölött helyezkednek el.

A 3A. és 3B. ábra együttes folyamatlistája a jelen találmányban hasznos biztonság kihelyezési protokoll működését ábrázolja. E szerint a protokoll szerint a 10' kliens két (2) különálló összeköttetést állít be minden egyes alkalommal,

amikor kapcsolatot kíván felépíteni egy adott forrás szerverrel. Egy első biztonságos összeköttetést állítunk fel a 10' kliens és a 15 proxy között, és ezt az összeköttetést használjuk csőként vagy csatornaként titkos információ átadására a kliens és a proxy között. Az első biztonságos összeköttetést a folyamatábra első két oszlopa ábrázolja. A 10' kliens továbbá felállít egy második biztonságos összeköttetést is a proxyval, amint azt a folyamatábra utolsó három oszlopa ábrázolja, viszont ebben az összeköttetésben a 15 proxyt alagútként használjuk a 12' forrás szerverhez. Egy alagút egy közvetítő program, amely vak reléként (blind relay) működik két kapcsolat között. Ha egy alagutat aktiválunk, nem tekintjük egy adott kommunikációban (például egy HTTP kérésben vagy válaszban) résztvevő félnek, bár az alagutat esetleg ez a kommunikáció kezdeményezte.

Ebben a bemutató jellegű példában feltesszük, hogy a kliens hozzá szeretne férni egy forrás szerverhez (amelyre néha első szerverként hivatkozunk) adott tartalom megszerzése érdekében, de a proxyt szeretné használni ennek a tartalomnak a megfelelő megjelenítéséhez. A kérés kiszolgálása megkívánhatja adott objektumok megszerzését egy vagy több további forrás szervertől. Amint azt fent megjegyeztük, az SSL/TLS protokoll szerint a kliens rendelkezik egy forrás szerver által kibocsátott tanúsítvánnyal a kliens hitelesítése céljából a forrás szerver felé, és a kliens is kéri a forrás szervert, hogy az mutasson be egy tanúsítványt, ami alapján a forrás szervert érvényesként hitelesítheti. Amint azt látni fogjuk, a kliens a

proxytól is megköveteli, hogy rendelkezzen egy tanúsítvánnyal, amit a kliens azelőtt hitelesít, mielőtt (a kliens) felfedne (a proxy számára) egy összeköttetés mester titkot.

A rutin a 20 lépésnél kezdődik, ahol a kliens egy biztonságos összeköttetést kér a proxyval. Ez lesz a fent azonosított első biztonságos összeköttetés. Amint az a folyamatábrán látható, a kliensnek muszáj egy tanúsítványt kérnie a proxytól, mivel éppen a saját biztonsági attribútumait szándékozik kihelyezni. Ez az elsődleges összeköttetés, amelyen keresztül a kliens küldeni fogja (a proxynak) bármely forrás szerver megbeszélt titkát, egy belső összeköttetés azonosítóval együtt. Ez az azonosító tipikusan nem azonos az SSL/TLS összeköttetés azonosítóval. Ezt részletesebben egy későbbi lépésben írjuk le.

A 22 lépésben a kliens hitelesíti a proxytól fogadott tanúsítvány érvényességét, és ennek eredményeképpen eleget teszünk annak, hogy egy biztonságos összeköttetéssel rendelkezzünk a proxyval. A rutin ezután a 24 lépésnél folytatódik, ahol a kliens egy második összeköttetést nyit meg a proxyhoz. Ez lesz a fent leírt második biztonságos összeköttetés. Amint azt megjegyeztük, a kliens kéri az egy adott forrás szerverhez való alagutazást (például a HTTP CONNECT eljárás használatával egy kérésnél). A proxyn keresztüli alagutazási kérés részeként a kliens hozzáfűz egy fejlécet a HTTP kéréshez, amelyben értesíti a proxyt, hogy egy belső összeköttetés azonosítót kellene generálnia. Ez a fejléc jelzi, hogy a kliens továbbítani szándékozza a mester titkot a proxynak egy jövőbeni alkalommal.

A 26 lépésnél a proxy egy egyedi belső összeköttetés azonosítót generál, és visszaküldi ezt az információt a kliensnek. A belső összeköttetés azonosító értékét csatoljuk a biztonságos HTTP válaszhoz. Ez az az érték, amelyet a kliens használni fog az összeköttetés mester titok proxyhoz való továbbításánál. A 28 lépésnél a proxy felépít egy kapcsolatot a forrás szerverrel, és megengedi adat áramlását a kliens és a forrás szerver között. Ennél a pontnál a proxy alagútként viselkedik. Nem válik „aktív proxyvá", amíg a kliens nem továbbítja az összeköttetés mester titkot, amint azt később látni fogjuk. A 30 lépésnél a kliens kézfogást hajt végre a forrás szerverrel egy összeköttetés mester titok megbeszélésére.

A rutin ezután a 32 lépésnél folytatódik. Ennél a pontnál a kliens elküldi (a proxynak) a belső összeköttetés azonosítót az összeköttetés mester titokkal együtt. Ezt az információt az elsődleges összeköttetésen küldjük, amint azt ábrázoljuk. A 34 lépésnél a proxy fogadja a belső összeköttetés azonosítót és az összeköttetés mester titkot. Ezt az információt használja a szükséges kriptográfiai információ előállítására, amelyet forrás szerver válaszok visszafejtésére, a szolgáltatott tartalom módosítására és/vagy adatnak a klienshez való küldését megelőző titkosítására használ. A proxy ezután átkapcsol „aktív proxyra" a forrás szerverrel való aktuális kapcsolathoz.

A 36 lépésnél a kliens elküld a forrás szerveren lévő valamely erőforrás iránti biztonságos HTTP kérést. Opcionálisan a 38 lépésnél a proxy visszafejtheti, és szükség szerint módosíthatja a kérést, és ezután titkosíthatja az új kérést és

küldheti el azt a forrás szerverhez. A 40 lépésnél a forrás szerver eleget tesz a kérésnek, és válasz adatot küld vissza a proxynak. Megjegyezzük, hogy a forrás szerver előnyösen nincsen tudatában a proxy aktív részvételének. A 42 lépésben a proxy fogadja a tartalmat, visszafejti és módosítja az adatot a kliens átkódolási igényeinek kielégítésére. Opcionálisan a 44 lépésnél a proxy további kapcsolatokat építhet fel a forrás szerverrel (ha a forrás szerver támogatja az összeköttetés-visszaállítást) abból a célból, hogy további adatokat szerezzen meg és/vagy javítsa a teljesítményt. Ha több kapcsolat van felépítve, titkosító blokk láncolást (CBC - cipher block chaining) alkalmazunk a titkosító beállítására. Ha a proxy nem épít fel további kapcsolatokat ennek az összeköttetésnek a részeként, értesítenie kell a klienst a titkosító specifikációs változásokról azáltal, hogy értesítést küld az elsődleges összeköttetésen keresztül az összeköttetés azonosítóval együtt. Ezt a folyamatot mutatjuk be a 46 lépésnél, és ez szükséges ahhoz, hogy lehetővé tegyük a kliens számára, hogy egy későbbi időpontban visszaállítsa ezt az összeköttetést a forrás szerverrel.

A jelen találmány szerint a proxynak szüksége lehet további biztonságos összeköttetésekre más forrás szerverekhez egy adott, a kezdeti forrás szerverhez való kliens oldali kérés feldolgozása közben. Ily módon, ha például a proxynak további biztonságos összeköttetésekre van szüksége más forrás szerverekhez, például az aktuális kérés átkódolásához, akkor egy értesítést küld a kliensnek, amelyben kéri a klienst, hogy

építsen fel egy új összeköttetést a szükséges minden egyes további forrás szerverrel. Ezt ábrázoljuk általánosan a 48 lépésnél. Végül az 50 lépésben a proxy titkosítja a végső átkódolt tartalmat, és elküldi a kliensnek.

A 4. ábra részletesebben mutatja be, hogy a proxy hogyan kezdeményez egy vagy több további biztonságos összeköttetést más forrás szerverekhez. Ebben a példában a 10'' kliens az átkódoló 15'' proxyval és a 12'' első forrás szerverrel az előzőekben leírt módon működik együtt. Amint azt az ábrán mutatjuk, az 1. Összeköttetés ábrázolja a kezdeti összeköttetést, amelyet a kliens épít fel a proxyval, és a 2. Összeköttetés az, ahol a kliens felépíti a biztonságos összeköttetést az első forrás szerverrel. Az ábrán bemutatott 1)-12) lépések megfelelnek a 3. ábra szerintieknek, tételesen a következők:

1) biztonságos összeköttetés létrehozása, tanusítvány küldés; 2) vett tanusítvány érvényesítése; 3) HTTP kapcsolás összeköttetés ID kéréssel; 4) összeköttetés ID hozzárendelés; 5) kapcsolat létesítés az első forrás szerverrel; 6) normál SSL/TLS handshake és mestertitok generálás; 7) proxy összeköttetés ID és mestertitok küldés; 8) proxy aktiválódás, átkódolás kezdete; 9) HTTP kérés küldés; 10) kliens módosításkérés küldés; 11) HTTP válasz és adat küldés; 12) új kapcsolat létesítés az aktuális összeköttetésen belül; 13) kliens forrás szerverrel való összekapcsolásának kérése.

Ha a (bemutató jellegű) átkódolási művelet közben a 15'' proxy úgy határoz, hogy szüksége van biztonságos adatra egy 17'' második forrás szervertől, a proxy megkéri a 10'' kli-

enst, hogy építsen fel egy második kapcsolatot a 17'' szerverrel, ismét a proxyn keresztüli alagutazással. Ez lehetővé teszi a kliens számára, hogy egy mester titkot alakítson ki a 17'' második szerverrel. Az utóbbi mester titokra néha második mester titokként hivatkozunk, megkülönböztetésként a kliensnek a proxyn keresztül az első forrás szerverhez való alagutazása eredményeként generált összeköttetés mester titoktól. A 4. ábrán lévő 13) lépés azt ábrázolja, hogy a 15'' proxy kérést intéz a 10'' klienshez. A 3)-7) lépéseket ezután megismételjük a 17'' második forrás szerverrel az előzőekben leírt módon.

A 15'' proxy rendelkezik azzal a képességgel, hogy különálló mester titkokat tartson fenn, ha szükséges, a biztonságos kommunikációkhoz egyrészről a kliens, másrészről pedig a vonatkozó forrás szerverek között. Ily módon a kliens és a proxy különálló mester titkokat tart fenn, egyet minden egyes forrás szerver összeköttetéshez, az eredeti kliens oldali kérés összefüggésén belül. Ez megengedi a proxy számára, hogy adatokhoz férjen hozzá és használja azokat a kliens nevében több forrás szerver esetén. Ha szükséges, a kliens eljuttathatja az összeköttetés mester titkot (titkokat) a proxyhoz ugyanazon a biztonságos összeköttetésen keresztül (például az 1. ábrán bemutatott 1. Összeköttetésen keresztül), illetve különböző biztonságos összeköttetések használatával.

Amint az látható, a kliens, egy adott forrás szerver és a proxy mind osztoznak egy mester összeköttetés titkon. Ha már egyszer a kliens és az adott forrás szerver megegyeztek egy mester összeköttetés titokban, akkor a titkot biztosítjuk a

proxyhoz egy olyan biztonságos összeköttetésen keresztül, amelyet megelőzően hoztunk létre a kliens és a proxy között. Másképpen fogalmazva, a kliens átadja (a proxynak) ezt a mester összeköttetés titkot azután, hogy felépítettük az elsődleges (vagyis az első) összeköttetést (a kliens és a proxy között). A forrás szervernek viszont nem kell tudatában lennie (és tipikusan nincs is) annak, hogy a proxy valamilyen munkát végez vagy más módon részt vesz a biztonságos kapcsolatban.

Mint látható, az ezen biztonság kihelyezés támogatásához szükséges változtatások minimálisak, és csak a klienst és a proxyt befolyásolják, azon forrás szervert (szervereket) nem, amely(ek) szükséges(ek) egy adott kliens oldali kérés feldolgozásához. Továbbá ez a folyamat nem követeli meg a klienstől, hogy felfedjen bármely információt a saját privát kulcsával kapcsolatban vagy azzal az eljárással kapcsolatban, amelyet a kliensnek a forrás szerver felé való hitelesítéséhez használ. Ezen kívül, mivel a kliens képes arra, hogy további kapcsolatokat építsen fel egy forrás szerverhez, megváltoztathatja a titkosító specifikációt vagy megszakíthatja az összeköttetést, így korlátozva a proxy képességét arra, hogy az más kapcsolatokat építsen fel a forrás szerverhez a kliens nevében.

Összefoglalva a kívánt változtatásokat, a kliensnek rendelkeznie kell azzal a képességgel, hogy fogadjon egy vagy több vonatkozó forrás szerverrel megbeszélt egy vagy több összeköttetés mester titkot, és hogy biztonságosan eljuttassa azokat a proxyhoz. A proxynak képesnek kell lennie a szükséges titkosítási információ előállítására a kliens mester titká-

ból/titkaiból, hogy lehetővé tegye a kliens összeköttetésében való részvétel elkezdését. A fenti eljárás nem követel meg semmiféle változtatást az összeköttetés titok megbeszélésénél használt kézfogási protokollban. A hozzáadódó teher a teljes hálózati forgalomra minimális, mivel csak egyetlen további összeköttetés jön létre a kliens és a proxy között, miközben a kliens szolgáltatásokat kér a proxytól. A forrás szerver(ek) esetén nincs szükség változtatásokra.

A kliens és a proxy közötti elsődleges összeköttetést tekinthetjük aszinkronnak, mivel minden egyes beérkező rekordhoz létezik egy összeköttetés azonosító. A klienstől a proxyra való írások (writes) függetlenül fordulnak elő a proxynak a kliensre való írásától, mert nem követelünk meg nyugtákat. Természetesen feltesszük, hogy az alul elhelyezkedő szállítási réteg megvalósít egy megbízható továbbítási eljárást. A proxy felöli kérések a klienshez új (további forrás szerverekhez való) kapcsolatok felépítésére előnyösen egy null* összeköttetés azonosítót használnak, mert egyet fogunk nekik megfeleltetni később a proxy által, amikor a kliens alagutazást kér. A teljesítménnyel kapcsolatos okokból a proxynak nem kell értesítenie a klienst a titkosító specifikációs változásokról, beleértve azt, hogy a kliensnek muszáj lesz a teljes hitelesítési kézfogást végrehajtania, mert nem lesz szinkronizációban az adott forrás szerverrel. Ez nagyobb terhet jelent a kliensen egy forrás szerverrel való kezdeti összeköttetés-felépítés közben, viszont csökkenti a felesleges beszélgetést (chatter), ha a proxy új kapcsolatokat épít fel, illetve ha további kéré-

seket küld az adott forrás szerverhez vagy valamely másik forrás szerverhez.

Számos alkalmazás létezik a proxy számára. A következőkben néhány jellegzetes példát mutatunk be.

A proxy egy ilyen használata az, hogy csökkentjük egy kliens számára a titkosítás/visszafejtés végrehajtásához szükséges számítási teljesítményt. Ha például a kliens egy tűzfal mögött helyezkedik el, a proxy használatával a kliensnek a hitelesítési lépéseket csak egyszer kell végrehajtania, és azután tulajdonképpen szabadon küldhet és fogadhat adatot közte és a proxy között, így a titkosítás terhét a proxyra helyezzük át. Alternatívaként a proxyt használjuk auditálási képességek biztosítására egy tűzfalas konfigurációhoz azáltal, hogy lehetővé tesszük a kliens számára (vagy megköveteljük a klienstől), hogy átadja az összeköttetés titkot, mielőtt bármely tényleges adat rekordot cserélhetne a forrás szerverrel. Ebben az esetben a proxynak nem kell a klienst arra kérnie, hogy az eljuttasson bármely privát/magán jellegű információt magáról vagy a forrás szerverről. Egy még további példában a proxyt használhatjuk a kliens teljesítményének javítására azáltal, hogy lehetővé tesszük egy ideiglenesen tároló proxy (caching proxy) részvételét az összeköttetésben, a kliens és a forrás szerver közötti összeköttetés biztonsági tulajdonságainak megváltoztatása nélkül. Alternatívaként a proxyt használhatjuk arra, hogy előre behozzon (pre-fetch) tartalmat a kliens nevében (az összeköttetések visszaállításával egy későbbi időpontban) anélkül, hogy a proxy explicit ismeretekkel rendelkezne a

kliens privát kulcsáról. Ebben az esetben a proxy megszerezhetné például a kliens előfizetéseinek rendszeres frissítéseit a csúcsforgalmi időn kívüli órákban. Ezek a példák pusztán be-

mutató jellegűek, és nem tekintendők a jelen találmány tárgykörét korlátozókként.

Ily módon, amint azt fent megjegyeztük, a jelen találmány egy másik alkalmazása az lehet, hogy lehetővé tesszük egy harmadik fél számára, hogy részt vegyen egy átjárható számítási kliens eszközt magában foglaló biztonságos összeköttetésben. A jellegzetes eszközök magukban foglalnak egy x86-, PowerPC®- vagy RISC-alapú átjárható klienst, valós idejű operációs rendszerrel, mint amilyen a WindRiver VXWorks™, QSSL QNXNeutrino™ vagy Microsoft Windows CE, és tartalmazhat egy Web böngészőt. Ezt az alkalmazást írjuk le alább részletesen.

Az 5. ábrára hivatkozva, egy jellegzetes átjárható számítási eszköz magában foglal egy 140 kliens vermet, amely számos komponenst tartalmaz, például egy 142 kliens alkalmazáskeretet, egy 144 virtuális gépet, egy 146 beszéd motort és egy iparág által biztosított 148 futási idejű operációs rendszert (RTOS - runtime operating system). A 142 kliens alkalmazáskeret tipikusan tartalmaz egy 150 böngészőt, egy 152 felhasználó interfészt, egy 154 átjárható számítási kliens alkalmazás osztály könyvtárat, egy 156 szabványos Java osztály könyvtárat és egy 158 kommunikációs vermet. Az átjárható számítási kliens egy 160 szerver platformhoz kapcsolódik egy 162 összeköttetési (connectivity) szolgáltatáson keresztül.

Az alsó szintjén a 162 összeköttetési szolgáltatás magában foglal egy 164 átjárót, amely tömörítési és titkosítási funkciókat biztosít. Az átjáró egy olyan hálózati biztonsági protokollt valósít meg, amelyet kiterjesztettünk a jelen ta-

lálmány eljárása szerint. A 162 összeköttetési szolgáltatás felső szintje a 166 proxy, amely egy vagy több különböző funkciót biztosít, mint például: átkódolást, szűrést, fontossági sorrendbe való állítást (priorizációt) és kapcsolatot az eszköz kezeléshez.

A 160 szerver platform, nevezetesen egy adott forrás szerver, számos, különböző típusú lehet. A 160 platform lehet egy 170 Web/alkalmazás szerver (egy szinkron kérés-válasz típusú szerver) vagy egy 172 adat szinkronizációs szerver (egy aszinkron soros kommunikáció típusú szerver). Az ilyen szerver típusok mindegyikének alap funkcióit mutatjuk be. Alternatívaként a 160 platform lehet egy 174 értéknövelt szerver, amely további szolgáltatásokat is biztosít, például LDAP (Lightweight Directory Access Protocoll, könnyített könyvtárhozzáférési protokoll) könyvtárat/tárolót, megfigyelést és értesítést, hálózat kezelést, eszköz életciklus kezelést, felhasználó- és eszközregisztrációt illetve számlázást.

A biztonság kihelyezési protokoll számos előnyt biztosít az eddig használatos megközelítéshez képest. Amint azt fentebb tárgyaltuk, a protokoll kiterjesztés nem változtatja meg egy biztonságos kapcsolat alap tulajdonságait a rekord protokoll rétegnél. Ezen túlmenően a proxyval való kapcsolat privát, és az adat titkosításhoz szimmetrikus titkosítást (például DES-t, RC4-et stb.) használhatunk. Ehhez a szimmetrikus titkosításhoz a kulcsokat előnyösen egyedien generáljuk minden egyes kapcsolat számára, és a kulcsok egy másik protokoll (például a TLS vagy az SSL kézfogási protokollok) által megbeszélt valamely

titkon alapulnak. Továbbá a proxyhoz való kapcsolat megbízható. Az üzenet szállítás tipikusan magában foglal egy kulcsos MAC-et használó üzenet integritás ellenőrzést. A MAC számításokhoz előnyösen biztonságos hasító függvényeket (például SHA-t, MD5-öt stb.) használunk.

A kézfogási protokoll biztosítja az összeköttetés biztonságot néhány alapvető tulajdonsággal. Az egyed azonosságát hitelesíthetjük aszimmetrikus, vagyis nyilvános kulcsú titkosítás (például RSA, DSS stb.) használatával. Ezt a hitelesítést opcionálisan végezhetjük el, de általában megköveteljük a felek legalább egyikénél. Ezen kívül egy megosztott titok megbeszélése is biztonságos. A megbeszélt titok nem elérhető a hallgatózók számára, és bármely hitelesített kapcsolat esetén a titok nem szerezhető meg, még egy olyan támadó által sem, aki képes magát a kommunikáció közepébe helyezni. Továbbá a proxyval való megbeszélés is megbízható. Egyetlen támadó sem módosíthatja a megbeszélt kommunikációt anélkül, hogy a kommunikációban részt vevő felek ne észlelnék.

Amint azt a fentiekben részletesebben tárgyaltuk, a biztonsági protokoll lehetővé teszi egy proxy számára, hogy részt vegyen egy kliens és forrás szerverek egy halmaza közötti valamely biztonságos összeköttetésben anélkül, hogy megváltoztatná az összeköttetés attribútumait. Az eljárás továbbá független a titkosítás erősségétől vagy a használt hitelesítési technikáktól.

A találmány megvalósítható processzorban végrehajtható szoftverként, nevezetesen utasítások valamely halmazaként

(program kódként) egy kód modulban, amely a számítógép közvetlen hozzáférésű memóriájában helyezkedik el. Amíg a számítógépnek nincs rá szüksége, az utasítások halmazát tárolhatjuk egy másik számítógépes memóriában, például egy merevlemezmeghajtóban vagy egy eltávolítható memóriában, illetve letölthető az Interneten vagy más számítógépes hálózaton keresztül.

Ezen túlmenően, bár a leírt változatos eljárásokat kényelmesen egy szelektív módon aktivált vagy szoftver által újrakonfigurált, általános célú számítógépben valósítjuk meg, a tárgykörben járatosak felismerhetik, hogy ezek az eljárások kivitelezhetőek hardveresen, köztes szoftverként (firmware), illetve valamely speciálisabb berendezésben, amelyet a kívánt eljárási lépések végrehajtására konstruáltak.

Szabadalmi igénypontok

1. Eljárás annak lehetővé tételére proxy (15) számára, hogy részt vegyen kliens (10') és első forrás szerver (12a') közötti biztonságos kommunikációban, **azzal jellemezve**, hogy magában foglalja a következő lépéseket:

(a) első biztonságos összeköttetést építünk fel (20, 22) a kliens és a proxy között;

(b) az első biztonságos összeköttetés ellenőrzése után második biztonságos összeköttetést építünk fel (24, 26) a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön az első forrás szerverhez;

(c) a kliens és az első forrás szerver megbeszélnek (30) egy összeköttetés mester titkot;

(d) a kliens eljuttatja (32) az összeköttetés mester titkot a proxyhoz az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt vegyen a biztonságos kommunikációban;

(e) válaszként kliens oldali kérésre az első forrás szerverhez, harmadik biztonságos összeköttetést építünk fel a kliens és a proxy között, amely harmadik biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön a második forrás szerverhez;

(f) a kliens és a második forrás szerver megbeszélnek új összeköttetés mester titkot; és

(g) a kliens eljuttatja az új összeköttetés mester titkot a proxynak az (a) lépésben generált első biztonságos összeköttetés használatával.

2. Az 1. igénypont szerinti eljárás, **azzal jellemezve,** hogy magában foglalja továbbá azt a lépést, hogy a proxy (15) az összeköttetés mester titkot és az új összeköttetés mester titkot használja adott kriptográfiai információ generálására.

3. A 2. igénypont szerinti eljárás, **azzal jellemezve,** hogy magában foglalja ezen kívül azt a lépést, hogy a proxy (15) aktív működési állapotba lép az összeköttetés mester titok (d) lépésben való fogadását követően.

4. A 3. igénypont szerinti eljárás, **azzal jellemezve,** hogy a proxy (15) adott szolgáltatást végez el a kliens (10') nevében az aktív működési állapotban.

5. A 4. igénypont szerinti eljárás, **azzal jellemezve,** hogy az adott szolgáltatás átkódolás.

6. Az 1. igénypont szerinti eljárás, **azzal jellemezve,** hogy az első és második összeköttetés megfelel hálózati biztonsági protokollnak.

7. Az 1. igénypont szerinti eljárás, **azzal jellemezve,** hogy a szerver (12a') Web szerver és a kliens (10') átjárható számítási kliens.

8. Eljárás annak lehetővé tételére proxy (15) számára, hogy részt vegyen kliens (10') és szerver (12a') közötti biztonságos kommunikációban, amely eljárást kliensben használjuk 1-„n" szerverek halmazával való kommunikáláshoz, **azzal jellemezve,** hogy magában foglalja a következő lépéseket:

(a) 1-„n" szerverek halmazának mindegyike esetén:

(1) a kliens első biztonságos kapcsolatot kér (20, 22) a proxyhoz;

(2) a proxytól fogadott tanúsítvány érvényességének hitelesítése után a kliens második biztonságos kapcsolatot kér (24, 26) a proxyhoz, amely második biztonságos kapcsolat kéri a proxyt, hogy csatornaként működjön a szerverhez;

(3) a kliens és a szerver megbeszél (30) vonatkozó összeköttetés mester titkot a csatornán keresztül;

(4) a megbeszélés befejezése után a kliens eljuttatja (32) a vonatkozó összeköttetés mester titkokat a proxynak az első biztonságos kapcsolat használatával; és

(b) a proxy a vonatkozó összeköttetés mester titkokat használja adott kriptográfiai információ generálására, amely hasznos a biztonságos kapcsolatban való részvételhez.

9. A 8. igénypont szerinti eljárás, **azzal jellemezve,** hogy magában foglalja továbbá azt a lépést, hogy a proxy (15) adott szolgáltatást végez el a kliens nevében, amely szolgáltatást szolgáltatások olyan halmazából választjuk ki, ami magában foglal átkódolást, ideiglenes tárolást, titkosítást, visszafejtést, monitorozást, szűrést és előre behozást.

10. Eljárás annak lehetővé tételére proxy (15) számára, hogy részt vegyen biztonságos kommunikációban, amely eljárást kliensben használjuk, **azzal jellemezve,** hogy magában foglalja a következő lépéseket:

(a) kérést továbbítunk (20) klienstől a proxyhoz első biztonságos összeköttetés felépítésére;

(b) kérést továbbítunk (24) a klienstől a proxyhoz második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön forrás szerverhez;

(c) összeköttetés mester titkot továbbítunk (32) a klienstől a proxyhoz az első biztonságos összeköttetés használatával annak lehetővé tételére a proxy számára, hogy részt vegyen a biztonságos kommunikációban;

(d) válaszként a proxytól származó kérés fogadására a kliensnél, kérést továbbítunk a klienstől a proxyhoz harmadik biztonságos összeköttetés felépítésére a kliens és a proxy között, amely harmadik biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön másik forrás szerverhez; és

(e) új összeköttetés mester titkot továbbítunk a klienstől a proxyhoz.

11. A 10. igénypont szerinti eljárás, **azzal jellemezve,** hogy az új összeköttetés mester titkot az első biztonságos összeköttetésen keresztül továbbítjuk.

12. Eljárás annak lehetővé tételére proxy (15) számára, hogy részt vegyen biztonságos kommunikációban, amely eljárást proxyban használjuk, **azzal jellemezve**, hogy magában foglalja a következő lépéseket:

(a) klienstől származó kérést fogadunk (20) a proxynál első biztonságos összeköttetés felépítésére a kliens és a proxy között;

(b) a klienstől származó kérést fogadunk (24) a proxynál második biztonságos összeköttetés felépítésére a kliens és a

proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön első forrás szerverhez;

(c) a klienstől az első biztonságos összeköttetés használatával továbbított összeköttetés mester titkot fogadunk (32) a proxynál;

(d) adott kérés továbbítása után a proxytól a klienshez, a klienstől származó kérést fogadunk a proxynál második biztonságos összeköttetés felépítésére a kliens és a proxy között, amely második biztonságos összeköttetés kéri a proxyt, hogy csatornaként működjön második forrás szerverhez; és

(e) a proxynál a klienstől továbbított új összeköttetés mester titkot fogadunk.

13. A 12. igénypont szerinti eljárás, **azzal jellemezve**, hogy magában foglalja továbbá azt a lépést, hogy a proxy (15) az összeköttetés mester titkot és az új összeköttetés mester titkot használja adott kriptográfiai információ generálására.

14. Eljárás annak lehetővé tételére proxy (15) számára, hogy kliens (10') és első forrás szerver (12a') közötti összeköttetésben vegyen részt, amely eljárást proxyban használjuk, **azzal jellemezve**, hogy magában foglalja a következő lépéseket:

klienstől származó kérést fogadunk (20) a proxynál első biztonságos összeköttetés felépítésére a kliens és a proxy között;

a proxyn keresztül biztonsági kézfogási eljárást vezetünk (20, 22, 24, 26, 30) a kliens és az első forrás szerver között első összeköttetés kulcs létrehozására;

a kliens továbbítja az első összeköttetés kulcsot a proxyhoz, ami a proxy részt vehet a kliens és az első forrás szerver közötti kommunikációkban az összeköttetés közben; és

az összeköttetés folytatásában biztonsági kézfogási eljárást vezetünk a kliens és második forrás szerver között, második összeköttetés kulcs létrehozására; valamint

a kliens továbbítja a második összeköttetés kulcsot a proxynak, ami által a proxy adatot szerezhet meg a második forrás szervertől, a klienstől az első forrás szerverhez való kérés kiszolgálásakor való használathoz.

15. A 14. igénypont szerinti eljárás, **azzal jellemezve,** hogy minden egyes összeköttetés kulcsot különböző biztonságos kapcsolaton keresztül továbbítunk a klienstől (10') a proxyhoz (15).

16. A 14. igénypont szerinti eljárás, **azzal jellemezve,** hogy minden egyes összeköttetés kulcsot ugyanazon a biztonságos kapcsolaton keresztül továbbítunk a klienstől (10') a proxyhoz (15).

17. Kriptográfiai rendszer, **azzal jellemezve,** hogy magában foglal:

klienst (10');

szerverek halmazát (12a', 12b');

proxyt (15);

hálózati protokoll szolgáltatást annak lehetővé tételére a kliens és minden egyes szerver számára, hogy biztonságos kapcsolaton keresztül kommunikáljanak;

számítógépes programot (i) a kliens vezérlésére, hogy első biztonságos kapcsolatot kérjen a proxyhoz, (ii) válaszként a proxytól származó tanúsítvány érvényességének hitelesítésére, a kliens vezérlésére, hogy második biztonságos kapcsolatot kérjen a proxyhoz, amely második biztonságos kapcsolat kéri a proxyt, hogy csatornaként működjön adott szerverhez, (iii) a kliens vezérlésére, hogy megbeszélést folytasson az adott szerverrel a csatornán keresztül összeköttetés mester titok megszerzésére, és (iv) a megbeszélés sikeres befejezése után, a kliens vezérlésére, hogy eljuttassa az összeköttetés mester titkot a proxynak az első biztonságos kapcsolat használatával; és

számítógépes programot (i) a proxy vezérlésére, hogy az összeköttetés mester titkot használja adott kriptográfiai információ generálására, (ii) a proxy vezérlésére, hogy kérje, hogy a kliens szelektív módon építsen fel különálló biztonságos kapcsolatot másik szerverrel, és (iii) a proxy átkapcsolására aktív működési állapotba, ami közben részt vehet a kliens és az adott szerver közötti kommunikációkban.

A meghatalmazott

Dr. Köteles Zoltán
szabadalmi ügyvivő
az S.B.G. & K. Szabadalmi Ügyvivői Iroda
tagja
H-1062 Budapest, Andrássy út 113.
Telefon: 461-1000 Fax: 461-1099

KÖZZÉTÉTELI PÉLDÁNY

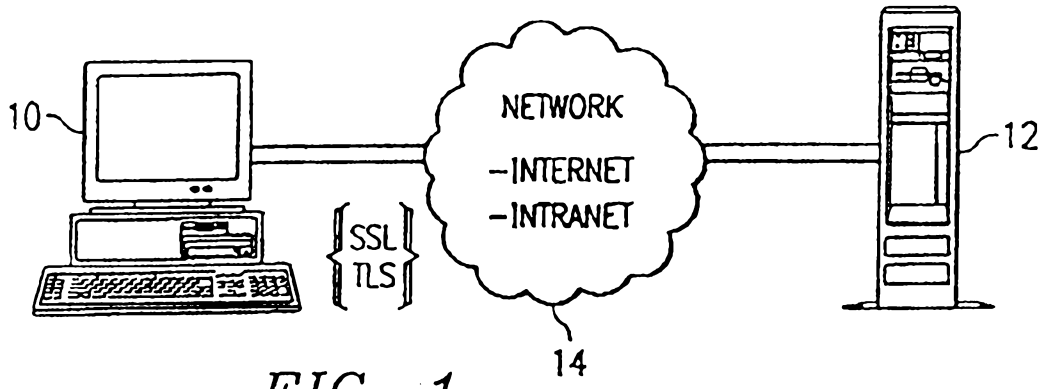


FIG. 1

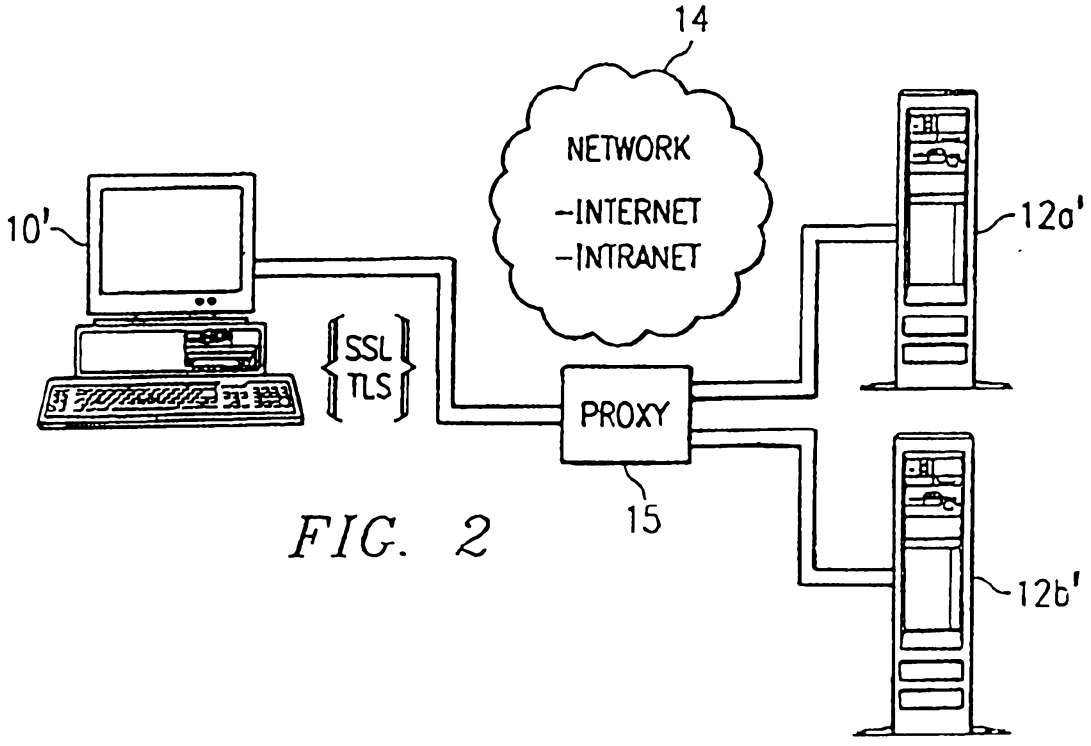
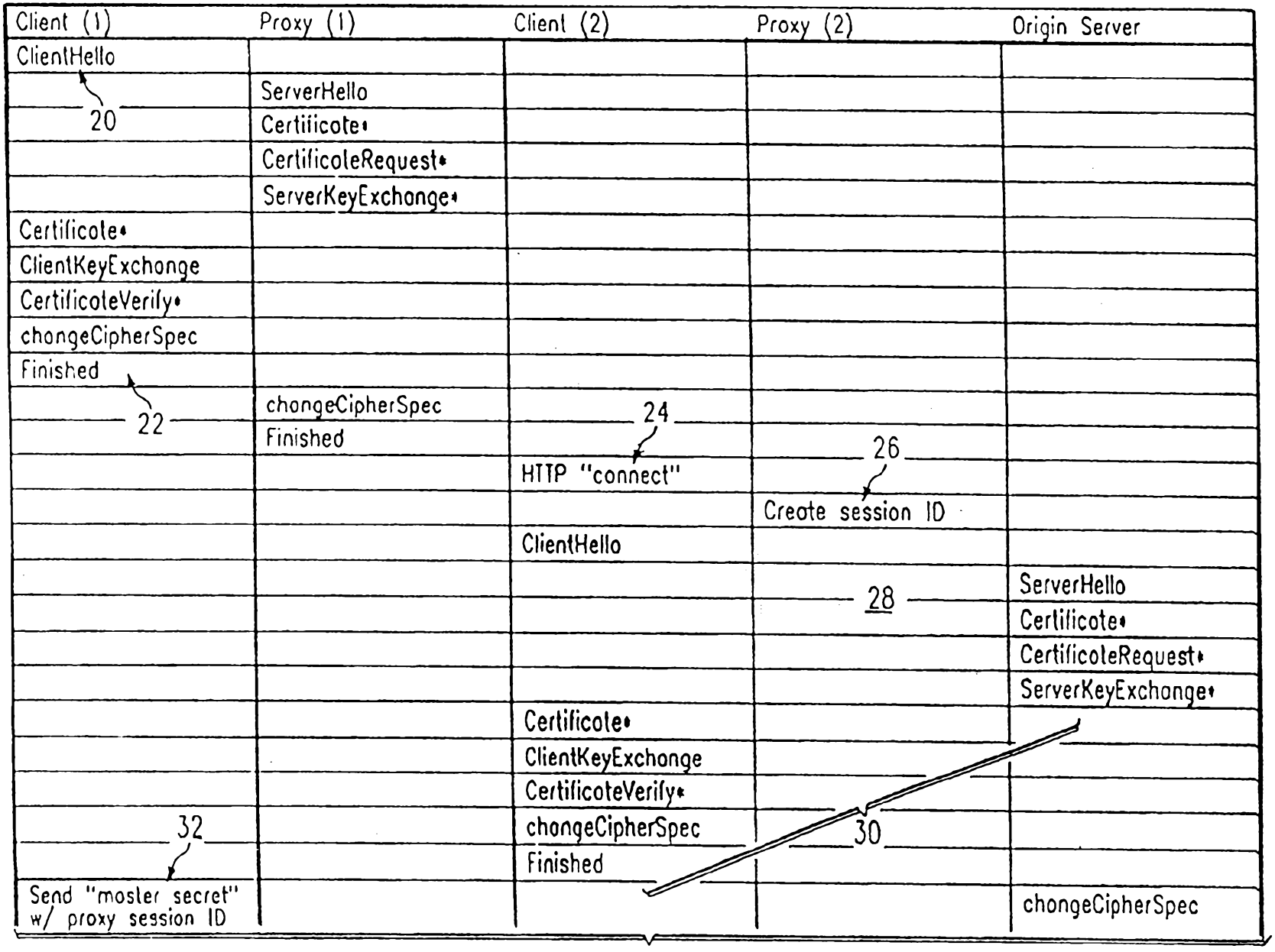


FIG. 2



TO FIG. 3B

FIG. 3A

FROM FIG. 3A

	Calculate key-block based on "moster secret"			Finished
	34	HTTP Request	38	
		36	Update/modify request	40
				HTTP content reply
			Transcode reply	
		42	Send second request	
				HTTP content reply
			(2) ClientHello. 44	
				(2) ServerHello
				(2) Certificate*
				(2) CertificateRequest*
				(2) ServerKeyExchonge*
			(2) Certificate*	
			(2) ClientKeyExchonge	
			(2) CertificateVerify*	
	46		(2) changeCipherSpec	
			(2) Finished	
	Notify client of cipher spec change			changeCipherSpec
changeCipherSpec			48	Finished
Finished			Send third request	
		50		HTTP content reply
			Transcode final reply	
		Render reply		

FIG. 3B

KÖZZÉTÉTELI PÉLDÁNY

3/5

P02017706

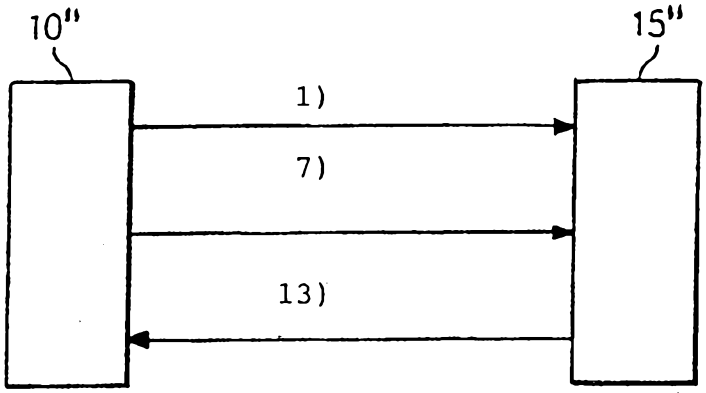


FIG. 4a

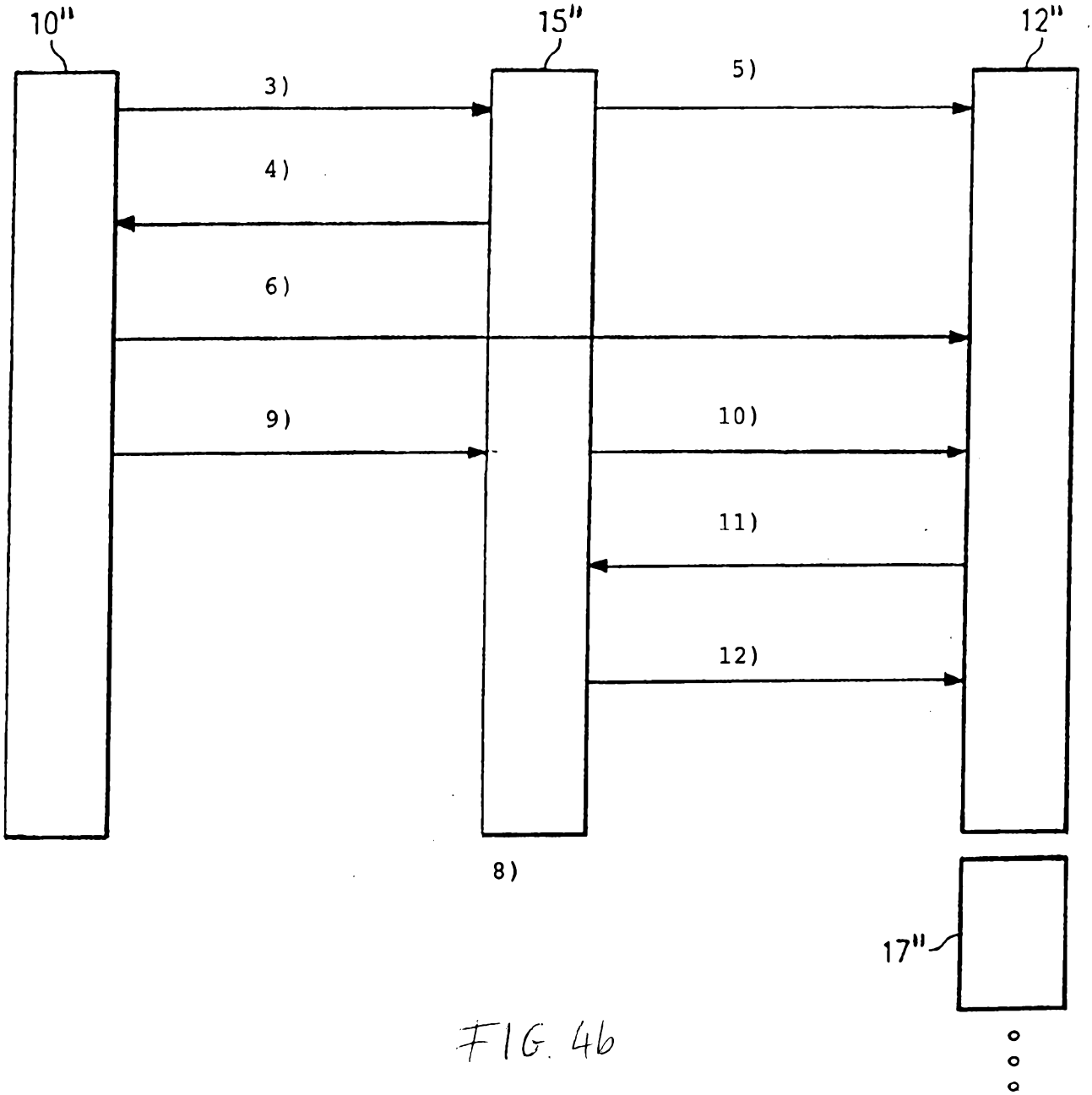


FIG. 4b

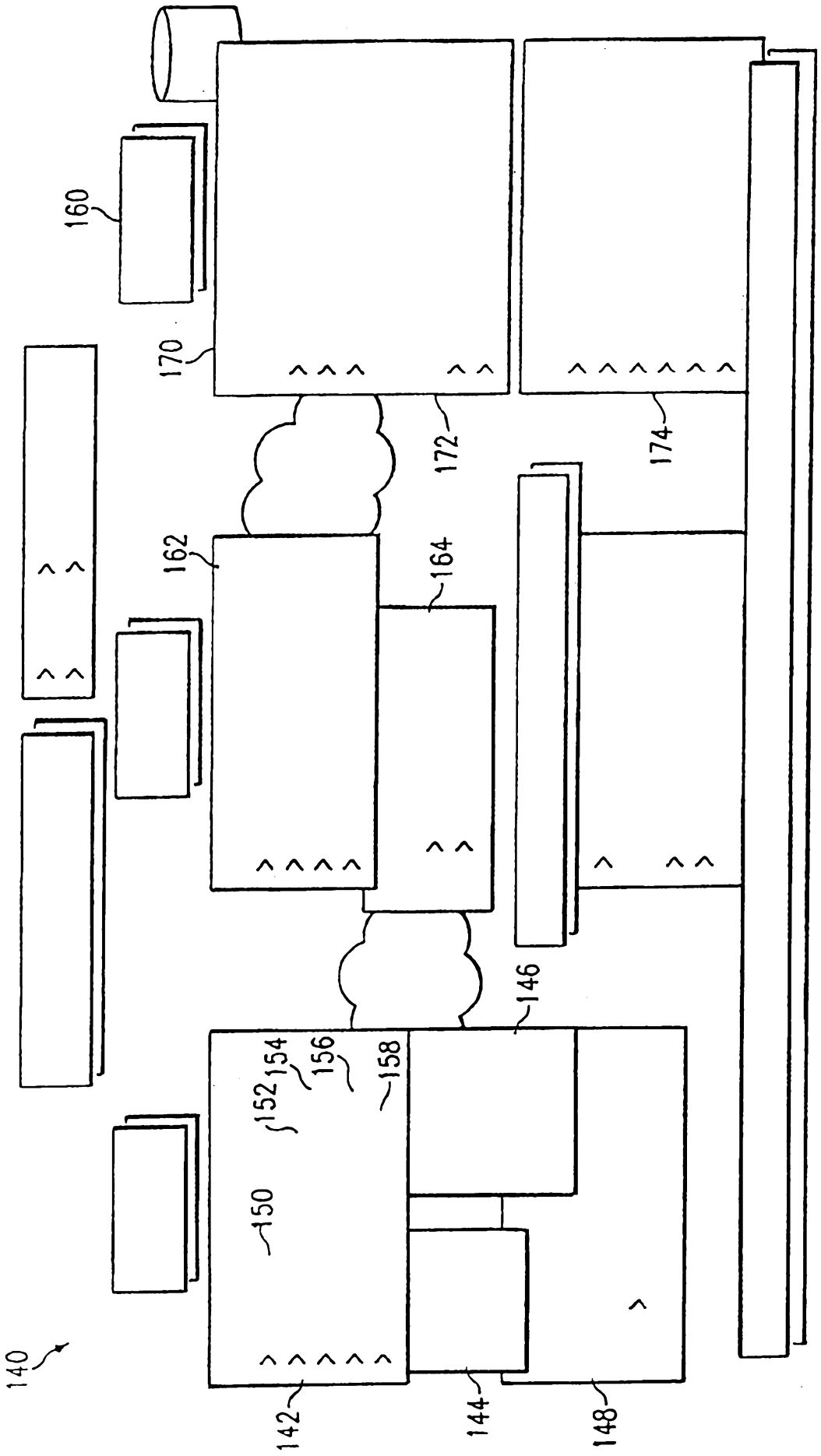


FIG. 5