Doktori (PhD) értekezés tézisei

Biztonságos Autentikációs Sémák Tervezése Elosztott Rendszerekre

Oláh Norbert

Témavezető: Dr. Pintér-Huszti Andrea

Debreceni Egyetem
Informatikai Tudományok Doktori Iskola

Debrecen, 2022

Összefoglaló

Az online kommunikáció során az egyik alapvető kérdés a résztvevők biztonságos hitelesítése. Ha a hitelesítés megfelelően működik, akkor elkerülhetőek a különböző támadások (pl. megszemélyesítéses támadás), ellenben az autentikáció helytelen működése esetén nem biztosított a felhasználó hozzáférés-ellenőrzés, illetve a felhasználói adatok bizalmassága és sértetlensége. A felhasználó hitelesítési sémák esetén számos biztonsági követelményt kell figyelembe venni, amelyek függenek az alkalmazott környezet jellemzőitől. Az egyik leggyakrabban használt hitelesítési módszer rövid titkokon, például jelszavakon alapul. Az első entitás hitelesítési fázis előtt mindig szükséges egy regisztrációs folyamat végrehajtása, mely a tudományos irodalomban kevés figyelmet kap.

A jelen disszertáció három új felhasználó hitelesítési protokollt, illetve egy felhasználói regisztrációs protokollt mutat be. Az autentikáció végrehajtása osztott, vagyis több résztvevő által történik a felhőalapú számítástechnikai szolgáltatások és az okos otthon környezetek magasabb biztonsági szintjének elérése érdekében. Formális elemzéssel bizonyítjuk, hogy a protokollok teljesítik a szükséges biztonsági követelményeket. Megoldásaink hatékonyabbak, mint a jelenlegi gyakorlati és elméleti sémák.

Az első fejezet a felhasználói hitelesítési rendszerek és megoldások

tudományos hátterét tartalmazza.

A második fejezetben részletezzük a protokolljainkban alkalmazott kriptográfiai primitíveket és megadjuk a szükséges definíciókat.

A 3. fejezet az automatizált biztonságelemző eszközökkel foglalkozik, és bemutatja a bizonyítható biztonság fogalmának részleteit.

A 4. fejezetben két, felhő környezetben alkalmazható elosztott felhasználó hitelesítési rendszert mutatunk be.

Az 5. fejezetben egy Identitás Alapú Kriptográfián és jelszón alapuló regisztrációs sémát ismertetünk, ahol a felhasználót és a szolgáltatót egyaránt hitelesíti a rövid életű, identitás alapú titkos kulcsa.

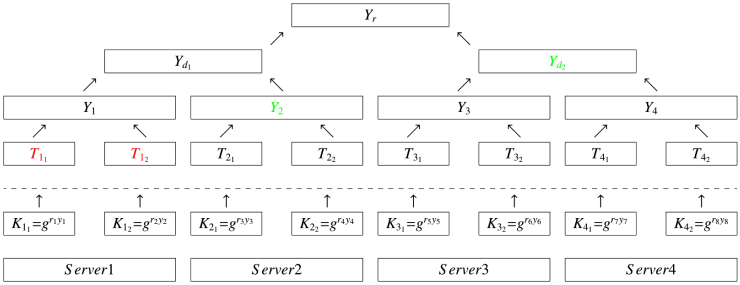
A 6. fejezetben bemutatunk egy küszöbszámon és jelszón alapuló, elosztott, kölcsönösen hitelesített kulcsmegegyezés és kulcskonfirmáció protokollt az okos otthon környezetekre.

Felhasználó hitelesítési protokoll Merkle-fa használatával

A 4.1. fejezetben bemutatunk egy felhő környezetben alkalmazható, Merkle fa használatán alapuló kétfaktoros hitelesítési sémát ([12]). Az elméleti ([8, 9]) és a gyakorlati megoldások centralizált hitelesítést alkalmaznak, ahol *egyetlen* felhőszerver végzi a felhasználók hitelesítését. A mi megoldásunk több szervert alkalmaz a felhasználók hitelesítésére. A [13] cikkben a séma biztonsági elemzését mutatjuk be applied pi kalkulusban. Protokollunkban a támadás csak akkor lehet sikeres, ha az ellenfél rendelkezik a szerverek által ismert összes jelszórésszel. A hitelesítési fázisunk hatékonyságát a [8, 9] munkájával összehasonlítva azt találjuk, hogy a mi sémánk hatékonyabb, mivel a résztvevő felek főleg csak hash számításokat végeznek. A fejezet eredményeit a Huszti Anderával közös cikkek ([12, 13]) tartalmazzák.

A felhasználó hitelesítése a szolgáltató oldalán egy statikus és egy egyszer használatos jelszóval történik egy véletlenszerűen kiválasztott szerver segítségével. A kiválasztott szerver a Merkle fát (1. ábra) alkalmazva az egyszer használatos jelszó helyességét tudja ellenőrizni. A Merkle fa levele egy jelszórész hash értéke, és

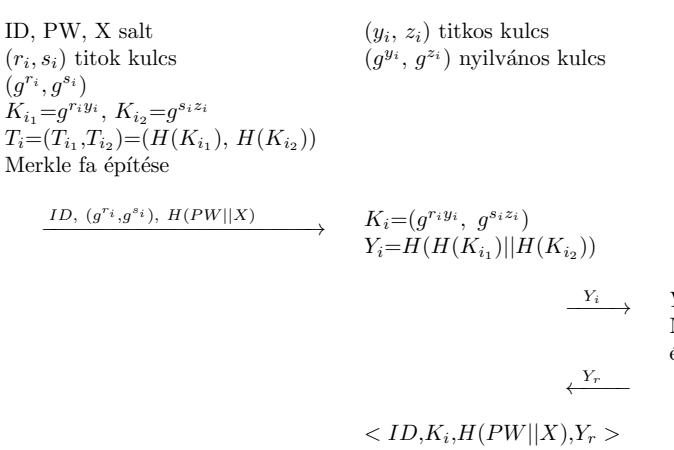
a fa gyökérelemével, illetve a hozzá tartozó Merkle fa útvonallal megtörténik a teljes egyszer használatos jelszó helyességének ellenőrzése.



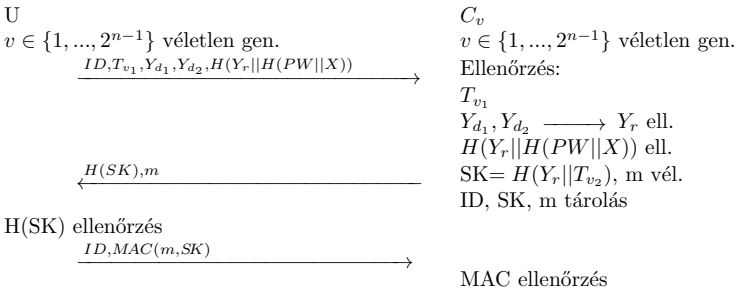
1. ábra. Merkle-fa 8 levélelemmel

A protokoll három fázisból áll: regisztráció, hitelesítés és szinkronizálás. A regisztrációs fázisban (2. ábra) a titkos kulcsok cseréje során nagy mennyiségű egyszer használatos jelszó generálódik a felhasználó és a felhőszerverek között. Minden felhőszerver (C_i) rendelkezik egy aszimmetrikus kulcspárral: $SK_{C_i} = (y_i, z_i)$, $PK_{C_i} = (g^{y_i}, g^{z_i})$, ahol g egy ciklikus csoport generátoreleme, és a $y_i, z_i \in \mathbb{Z}_q$ véletlen értékek, ahol q egy nagy prím. A hitelesítési fázisban (3. ábra) megtörténik a felhasználó és egy véletlenszerűen kiválasztott felhőszerver (C_v) kölcsönös hitelesítése, valamint végrehajtódik egy MAC kulcscsere. Üzenet hitelesítési kulcs (MAC) cseréje garantálja az üzenetek változatlanságát és eredetének integritását a későbbi interaktív kommunikáció során.

A hitelesítés után a szinkronizálás folyamata következik (4. ábra), ahol a kiválasztott szerver jelszava frissül a fához tartozó útvonallal együtt. A biztonsági elemzéshez a protokollt ProVerifben formalizáljuk. A ProVerif teszt eredménye azt mutatja, hogy a megadott biztonsági kritériumok teljesülnek. Ezt a protokollt kulcscsere sémaként elemezzük, így figyelembe vesszük a kölcsönös en-



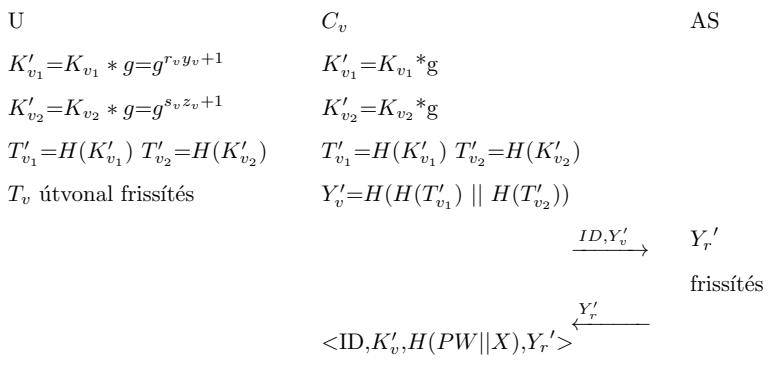
2. ábra. Regisztráció



3. ábra. Autentikáció

titás hitelesítési sémák tipikus biztonsági követelményeit, valamint a kulcsokkal kapcsolatos követelményeket. A következő négy tulajdonságot sikerült igazolni:

1. Kölcsönös hitelesítés



4. ábra. Szinkronizáció

(a) A felhasználók hitelesítése: A támadók nem adhatják ki magukat legális felhasználónak, és nem férhetnek hozzá a felhasználói adatokhoz.

(b) A szerver hitelesítése: A támadók nem adhatják ki magukat legális felhőszervernek.

2. A MAC kulcs titkossága: A kulcscsere során az újonnan generált kulcs bizalmas adat, és a támadónak nem szabad információval rendelkeznie az új kulcsról.

3. Kulcs frissessége: A protokoll futása közben egy új, véletlenszerűen kiválasztott kulcsot kell generálni, így a protokoll végrehajtása nem lehet sikeres egy régi, már korábban használt kulccsal.

4. Mindkét félnek ellenőriznie kell, hogy a másik fél ismeri és tudja használni az új MAC-kulcsot.

A felhasználó és a szerver hitelesítésének biztonsági elemzésére injektív lekérdezéseket alkalmazunk. A lekérdezések mindegyike igaz értékkel tér vissza, ami azt jelenti hogy a modellünkben a felhasználó

és a szerver kölcsönös hitelesítésének megsértésére, valamint a kulcs titkosságának sérülésére nem talál támadást a ProVerif. A kölcsönös hitelesítés mellett a *kulcs frissessége és kulcskonfirmáció* szempontok is teljesülnek.

Skálázható és elosztott felhasználó hitelesítés felhő szolgáltatásokhoz

A 4.2. fejezetben egy többszerveres jelszó alapú hitelesített kulcscsere sémát (5. - 7. ábrák) javasolunk. Más, küszöbszámon alapuló titokmegosztási algoritmusokat alkalmazó és jelszó alapú protokollokkal [1, 3, 6, 7, 23, 18, 17, 22, 19] ellentétben, habár a jelszóinformációkat megosztjuk a szerverek között, a titkot nem kell rekonstruálni a titokrészekből, hogy ellenőrizze a felhasználó hitelességét. Annak bizonyítására, hogy a javasolt protokoll bizonyíthatóan biztonságos, bevezetjük a küszöbszám alapú hibrid korrupciós modellt. A [6, 10]-tól eltérően részletes biztonsági elemzést adunk a Bellare és Rogaway modell alapján. Más sémákkal összehasonlítva figyelembe vesszük a skálázhatósági tulajdonságot is, amely az egyik fő követelmény a felhőkkel szemben és bemutatunk egy új módot arra, hogy a jelszóból skálázható erős titkot állítsunk elő (pl. hosszú élettartamú kulcsot). A [24] cikkben a szerzők IoT környezetbeli vezeték nélküli szenzorhálózatokra tervezett hitelesített kulcscsere (AKE) protokollt mutatnak be. A szerzők a kulcsmegosztásokra összpontosítottak, és egy hitelesített kulcscserét javasolnak a vezeték nélküli szenzorhálózat (WSN) és a központi hitelesítést végző felhőszerver között. Az AKE protokoll egy módosított változatát a [25] publikációban ismertetik, mely 5G hálózatra tervezett, és a felhőszerverek mellett egy rögzített vezérlőszervert tételez fel. A javaslatunk eltér ezektől a megoldásoktól ([25, 24]), mivel a generált hosszú élettartamú kulcsokat a felhasználói és a szolgáltatói oldalon is skálázhatjuk. A korábban javasolt protokollunkhoz ([13]) képest a skálázhatóság mellett titokmegosztási technikát alkalmazunk. A fejezet eredményeit Huszti Andreával közös [14] cikkünk tartalmazza.

Az általunk javasolt protokoll sikeres lefutása egy munkamenet-kulcsot eredményez, amely biztosítja a résztvevők közötti későbbi üzenetek bizalmasságát. A protokollnak két fázisa van. A regisztráció során a kliens jelszó alapú, hosszú élettartamú kulcsokat cserél ki az összes (n darab) szerverrel. A kliens oldalon egy egyszerű megoldást javasolunk, amelyben a kliens jelszóval fér hozzá a hosszú élettartamú kulcsokhoz. Feltételezzük, hogy a kliens egy klienseszközzel rendelkezik, melyen (pl. intelligens kártya, mobiltelefon stb.) fut egy kliensszoftver, amely jelszót kér a felhasználótól a hitelesítési folyamat elindításához. Miután a kliens megadta a jelszót, a kliensszoftver legenerálja a hosszú élettartamú kulcsokat, és megkezdődik a hitelesítés végrehajtása. A jelszó helyességét nem a kliensszoftver, hanem a szerver ellenőrzi, így a klienseszköz nem tárol semmilyen információt a jelszóról. A felhasználó n szerverből véletlenszerűen kiválaszt k szervert a hitelesítéshez. A hitelesítés során a szerver csak a szimmetrikus, hosszú élettartamú K_i kulcs ($i \in \{1, \ldots, k\}$) ismeretében tudja kiszámítani a kliens által generált w kihívásértéket. A *KKDF* egy Keyed Key Derivation Function-t jelöl, amely egy m és egy *key* üzenethez egy K titkos kulcsot generál. A hitelesítési szerver (J_v) a résztvevő szerverektől kapott összes k darab kihívás érték helyességének ellenőrzésével hitelesíti a klienst.

A javasolt protokollban a szerverek biztonságos csatornákon kommunikálnak egymással. Egy véletlenszerűen kiválasztott szerver kommunikál a klienssel, így a kliensnek nem kell párhuzamosan kommunikálnia az összes k szerverrel és biztonságos csatornákat kiépíteni. A protokoll tervezése során a hitelesítés hatékonyságát MAC és egyéb gyorsnak számító kriptográfiai algoritmusok (hash, xor művelet, szimmetrikus titkosítás) biztosítják. A protokoll bizonyíthatóan biztonságos. Feltételezzük, hogy a támadó ($\mathcal{A}$) számára engedélyezett a `Send, Reveal, Corrupt, Test` lekérdezések végrehajtása.

Az elosztott hitelesítés elemzésére az alapmodellt a küszöbszám alapú hibrid korrupciós modellel bővítjük. Feltételezzük, hogy a résztvevők korruptak lehetnek. A modell *erős korrupciós modell* ([2]), ha a hosszú életű kulcsok $K_{I,J}$ és a résztvevő I által tárolt

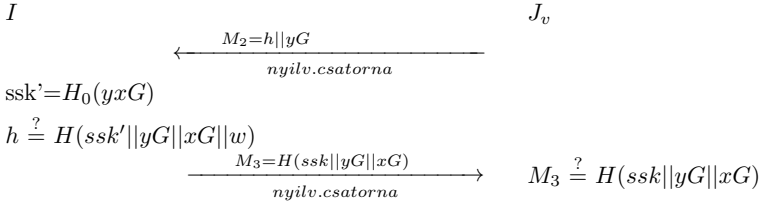
I J_v
$(K_1,\ldots,K_k)$, G K_v, G
$K_i = KKDF_{key}^{c+i}(psw)$, ahol $key = H(salt||psw)$
$K_n = KKDF_{key}(psw) \oplus \cdots \oplus KKDF^{c+n-2}{}_{key}(psw)$
$t_1,\ldots,t_{k-1},t_v$; $r_1,\ldots,r_{k-1}$, r_v, x véletlen
$w_1 = H(t_1),\ldots,w_v = H(t_v)$
$w = H(w_1||\ldots||w_{k-1}||w_v)$
$m_0 = H(w)$
$m_i = (MAC_{K_i}(r_i \oplus J_i) \oplus w_i)||r_i$
$m_v = (MAC_{K_v}(r_v \oplus xG \oplus J_v) \oplus w_v)||r_v||xG$
$\xrightarrow[nyilv.csatorna]{M_1=I||J_1||\ldots||J_k||m_0||\ldots||m_k}$

5. ábra. Hitelesítés - Kliens folyamat

J_v Szerverek
$\overline{K}_1,\ldots,\overline{K}_{k-1}$ rövid életű kulcsok
$\xrightarrow{I||m_i}$ J_i
K_i, $\overline{K}_i$
$m_i = p||o$
$w'_i = p \oplus MAC_{K_i}(o \oplus J_i)$
$\xleftarrow{Enc_{\overline{K}_i}\ (w'_i)}$
$m_v = u||s||z$
$w'_v = u \oplus MAC_{K_i}(s \oplus z \oplus J_v)$
$w' = H(w'_1||\ldots||w'_{k-1}||w'_v)$
$m_0 = H(w) \stackrel{?}{=} H(w')$
y véletlen
ssk=$H_0(yxG)$
$h = H(ssk||yG||xG||w')$

6. ábra. Hitelesítés - Felhő szerverek közötti kommunikáció

összes érték (pl. véletlenszerűen kiválasztott titkos értékek) a protokoll futása során az $\mathcal{A}$ támadó tudomására jut. A *gyenge korrupciós*



7. ábra. Hitelesítés - Végső folyamat

modell esetén csak a $K_{I,J}$ hosszú életű kulcsok módosulnak vagy kerülnek ki, a támadó nem kompromittálja teljesen a gépet. A protokollfutás során létrehozott és tárolt egyéb értékek nem kerülnek nyilvánosságra.

1. Definíció Egy modellt *küszöbszámon alapuló hibrid korrupciós modellnek* nevezünk, ha feltételezzük, hogy az autentikált kulcscsere kulcskonfirmációval protokoll (AKC) során n szerverből véletlenszerűen kiválasztunk k szervert, valamint a kliens nem korrupt, n szerver közül legalább $n-k+1$ szerver nem korrupt. Ezen kívül a klienssel való kommunikációnál a kiválasztott szerver

1. nem korrupt, vagy

2. gyengén korrupt, és a fennmaradó szerverek között van legalább egy nem korrupt.

A biztonságos AKC protokoll definíciójának megadásához át kell tekintenünk a [5] alapján a beszélgetés és az illeszkedő beszélgetés definícióját.

Az illeszkedő beszélgetés az I és a J entitások közötti valós idejű kommunikációt formalizálja. A $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ esemény definíciója a [5] dolgozatban megadott definíció módosított változata. Többszerveres beállításunkban minden kliens kommunikálhat gyengén korrupt szerverrel feltéve, hogy van legalább egy nem korrupt szerver a k szerverek között.

2. Definíció A P protokollban a $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ egy olyan esemény, ahol egy $\mathcal{A}$ támadó jelenlétében küszöbszám alapú hibrid korrupciós modellt tételezünk fel és létezik

1. egy $\prod_{I,J}^{s}$ kliens orákulum, mely elfogadott állapotban van, de nincs $\prod_{J,I}^{t}$ szerver orákulum, amely illeszkedő beszélgetést folytatna $\prod_{I,J}^{s}$ orákulummal, vagy

2. egy $\prod_{I,J}^{s}$ szerver orákulum, amely nem korrupt és elfogadott, de nincs olyan kliens orákulum, amelyik $\prod_{J,I}^{t}$ illeszkedő beszélgetést folytatna a $\prod_{I,J}^{s}$-vel, vagy

3. egy $\prod_{I,J}^{s}$ szerver orákulum, amely gyengén korrupt és elfogadott, de nincs kliens vagy nem korrupt szerver orákulum, amely illeszkedő beszélgetést folytatna a $\prod_{I,J}^{s}$-val.

A biztonságos AKC meghatározásához szükséges a *frissesség* fogalmának meghatározása és az *jóindulatú támadó* újradefiniálása.

3. Definíció Egy klienst és k szerver orákulumot tartalmazó elem $k+1$-es friss, ha a küszöbszámon alapuló hibrid korrupciós modellben a kliens orákulum és a szerver orákulum, amellyel illeszkedő beszélgetést folytatott, nem nyitott (unopened). Az orákulumot *frissnek* nevezzük, ha eleme egy friss elem $k+1$-esnek.

4. Definíció Egy támadót *jóindulatúnak* nevezünk, ha determinisztikus, és tevékenységét arra korlátozza, hogy választ egy elem $k+1$-es orákulumot, amely egy klienst és k szerver orákulumot tartalmaz, majd minden üzenetet tisztességesen továbbít egyik orákulumtól a másikig, a kliens orákulumtól indulva.

5. Definíció A protokoll egy *biztonságos AKC protokoll*, ha

1. A jóindulatú támadó jelenlétében a kliens és a klienssel kommunikáló szerver orákulum mindig elfogadja ugyanazt az ssk munkamenetkulcsot, mely egyenletes eloszlással generált a $\{0,1\}^{\kappa}$ halmazon.

minden $\mathcal{A}$ támadó jelenlétében

2. Egy küszöbszám alapú hibrid korrupciós modellben van egy kiválasztott $\prod_{I,J}^{l}$ szerver orákulum, amely illeszkedő beszélgetést folytat a kliens orákulummal, és ha ez a $\prod_{I,J}^{l}$ szerver orákulum gyengén korrupt, akkor a $\prod_{I,J}^{l}$ szerver orákulum illeszkedő beszélgetést kell folytatnia egy nem korrupt szerver orákulummal. A kliens orákulum és a $\prod_{I,J}^{l}$ szerver orákulum elfogadja és ugyanazt *ssk* munkamenetkulcsot használja.
3. A $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ valószínűsége elhanyagolható.
4. Ha a tesztelt orákulum friss, akkor $Adv^{\mathcal{A}}(\kappa)$ elhanyagolható.

1. Tétel. A javasolt protokoll egy biztonságos AKC protokoll a véletlenszerű orákulum modellben, feltételezve, hogy a MAC *univerzálisan hamisíthatatlan adaptív, választott üzenet alapú támadás esetén*, a szimmetrikus titkosítási séma *megkülönböztethetetlen a választott nyílt szöveg alapú támadásnál*, és az *Elliptikus görbe Diffie-Hellman kiszámíthatósági (ECCDH) probléma nehéz* az elliptikus görbe csoportban.

Protokollunk tervezése során fontos szempont volt a hatékonyság. A protokollban a munkamenetkulcsot az Elliptikus görbe Diffie-Hellman (ECDH) kulcscsere állítja elő, a többi művelet pedig a hash és xor műveletek, amelyek rendkívül gyorsak.

Bizonyíthatóan biztonságos identitás alapú távoli jelszó-regisztráció

Az 5. fejezetben egy Identitás Alapú Kriptográfián és jelszón alapuló regisztrációs sémát mutatunk be, ahol a felhasználót és a szolgáltatót egyaránt hitelesíti a rövid életű, identitás alapú titkos kulcsa. A javasolt protokoll illeszkedik az okos otthon környezetben alkalmazott felhasználó hitelesítési sémánkhoz, ahol a bilineáris leképezés értékeit az IoT-eszközökön tárolják. A javasolt felhősémánk ([13]) is könnyen módosítható a megfelelő hosszú élettartamú kulcsbeállítással, hogy kompatibilis legyen regisztrációs sémánkkal.

A biztonságos tároláshoz egy salt-tal ellátott bilineáris leképezést mutatunk be, ahol a salt egy rövid (12–48 bites) véletlenszerű adat, amelyet a hashelés előtt összefűznek a jelszóval. Így offline támadás esetén a támadó minden lehetséges jelszójelölthöz és salthoz kénytelen számításigényes bilineáris leképezést számolni, ami lassítja a támadást. A megoldásunk a hagyományos regisztrációs megoldásokkal ellentétben nem igényel Transport Layer Security (TLS) csatornát és mellőzi a hozzá tartozó tanúsítványkezelést is. Ez vállalati vagy oktatási intézményekben hatékonyabb működést tesz lehetővé, ahol jellemzően az egyedi azonosítók használata miatt ideális az Identitás Alapú Kriptográfia alkalmazása. A protokollunk hatékonyabb, mint a fent említett TLS-alapú és a többi vak regisztráció [20, 21], mivel nincs szükség tanúsítványok kezelésére vagy költséges nulla ismeretű bizonyítás végrehajtására. A többi rendszerrel ellentétben ([20, 21]) a jelszó hash-elő séma mellett figyelembe vettük az interakciókat is a protokoll résztvevői között és a jelszót ellenőrző információt biztonságosan küldjük. Bebizonyítottuk, hogy megoldásunk az online támadások ellen is biztonságos. Bevezetjük a biztonságos jelszóregisztációs rendszer definícióját, illetve megadjuk a támadói modellt és megmutatjuk, hogy a rendszerünk bizonyíthatóan biztonságos. A regisztrációnk rugalmas, ami optimális a föderációs bejelentkezésnél (SSO) vagy a Kerberos hitelesítéseknél, de olyan rendszereknél is alkalmas, ahol minden egyes szolgáltatáshoz más-más jelszót kell alkalmazni. A jelszó és a salt bilineáris leképezése hosszú élettartamú szimmetrikus kulcsként használható, és alkalmazható entitás hitelesítésre vagy munkamenetkulcs generálásra. A fejezet eredményeit az elfogadott cikkünk tartalmazza ([4]), amely Huszti Andreával, Bertók Csanáddal és Kovács Szabolccsal közös munka.

A protokoll egy beállítási és egy regisztrációs fázisból áll (8. ábra, 9. ábra). A Beállítás folyamata során legeneráljuk a rendszerparamétereket és a kulcsokat a résztvevők számára. Legyen P a $\mathbb{G}$ egy generátora, ahol $\mathbb{G}$ egy q-adrendű additív csoport, ahol q egy nagy prím. Válasszunk egy véletlen $\alpha \in \mathbb{Z}_q^*$ értéket, és generáljuk le a $P, \alpha P$ paramétereket. A rendszer mester titkos kulcsa az α. A ID_C,

ID_S azonosítók, a $PK_C = Q_C = tr(ID_C)$ és $PK_S = Q_S = tr(ID_S)$ nyilvános kulcsok. Mivel a jelszó hash sémánk elliptikus görbén alapuló bilineáris párosításokat ($\hat{e}$) használ, hatékony módszerre van szükségünk ahhoz, hogy a jelszavakat először egy $\mathbb{Z}_p$-beli elemre képezzük le, ahol p egy nagy prím, majd a $\mathbb{Z}_p$-beli elemet a görbe egy pontjához rendeljük hozzá. Jelöljük ezt a függvényt tr-rel. A privát kulcsgenerátor (PKG) kiszámítja a résztvevők titkos kulcsait ($SK_C = \alpha Q_C$ és $SK_S = \alpha Q_S$).

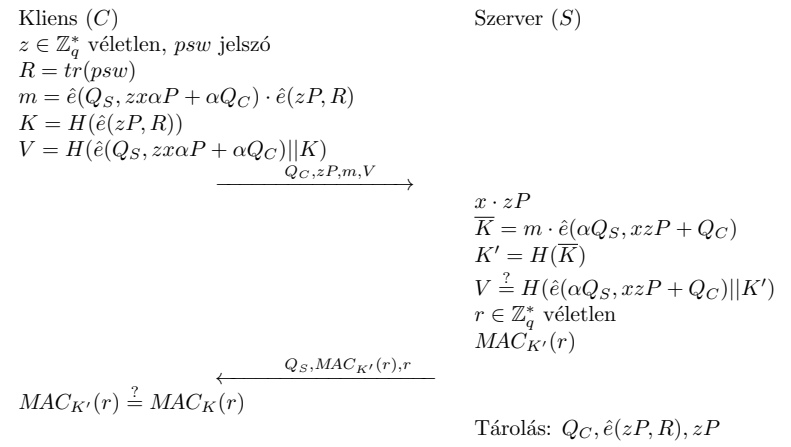
Kliens (C)	PKG	Szerver (S)
	$\alpha \in \mathbb{Z}_q^*$ (msk) nyilvános információk: $P, \alpha P, x\alpha P$	$x \in \mathbb{Z}_q^*$ titkos kulcs
$Q_C = tr(ID_C)$ (PK_C) αQ_C (SK_C)		$Q_S = tr(ID_S)(PK_S)$ αQ_S (SK_S)

8. ábra. Beállítás

A regisztrációs fázisban az ügyfelek elküldik jelszóadataikat a szervernek, és meggyőződnek arról, hogy a szerver megkapta a jelszóellenőrző értéket. A protokoll minden szükséges követelménynek megfelel, beleértve a jelszó titkosságával, a felek kölcsönös hitelesítésével és az offline támadásokkal szembeni ellenállást.

Olyan biztonsági modellt adunk meg, amely az offline támadások mellett az online támadásokkal szembeni ellenállást is tekinti. A javasolt modellünk a teljes regisztrációs folyamatot figyelembe veszi, ellentétben a [21] és [20] modellekkel. Tekintetbe veszi a kliens és a szerver közötti összes kommunikációs üzenetet. Ezért a résztvevők kölcsönös hitelesítését és a jelszó titkosságát is vizsgáljuk az átvitel során. Az $\mathcal{A}$ támadó olyan lekérdezéseket végezhet, amelyek modellezik a támadásait. Ezek a lekérdezések a következők: `Send`, `Corrupt`, `Reveal`, `Test`, `Execute` és `Finalise`.

Meghatározzuk a jelszóregisztrációs protokollok biztonsági céljait a teljes regisztrációs folyamatra vonatkozóan. Bevezetjük a biz-



9. ábra. Jelszó regisztrációs protokoll

tonságos regisztráció definícióját:

6. Definíció A protokoll egy *biztonságos regisztrációs protokoll* ha

1. A *jóindulatú támadó* jelenlétében a kliens és a vele kommunikáló szerver orákulum mindig elfogadott állapotba kerül. A szerver tárolja az ügyfél által megerősített jelszó ellenőrzési értéket.

és minden $\mathcal{A}$ támadóra

2. Ha van egy nem korrupt kliens orákulum, amely illeszkedő beszélgetéseket folytat egy nem korrupt szerver orákummal, akkor mindig elfogadott. A szerver tárolja a kliens által megerősített jelszó ellenőrzési értéket;

3. Nem korrupt szerver és kliens orákulum esetén a $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ valószínűsége elhanyagolható;

4. A tesztelt orákulumban a $Adv^{\mathcal{A}}(\kappa)$ elhanyagolható. Ha ez egy kliens orákulum, akkor nem nyitott;

5. Ha az összes D_n szótárnál az $\mathcal{A}$ támadó legfeljebb t darab (C, S, psw) elemhármast generál, akkor

$$Pr[\texttt{Finalise}(C, S, psw) = 1] \leq \frac{t}{2^{\beta_{D_n}} \cdot t_{pre}} + \mu(\kappa),$$

ahol $\mu(\kappa)$ elhanyagolható, a t_{pre} pedig az egyirányú függvény bemeneti értékének kiszámításához szükséges számítási költséget jelöli.

A protokoll biztonságát véletlen orákulum modellben vizsgáljuk, ahol két biztonsági modellt különböztetünk meg. A kliens-szerver protokollok esetében a kliensekről általában feltételezhető, hogy rosszindulatúak, azaz eltérnek a protokoll lépéseitől és bármilyen típusú stratégiát alkalmazhatnak a támadás során. A szolgáltatást nyújtó szerverek általában *becsületesnek* számítanak, vagyis nem indítanak támadást, vagy *becsületes, de kíváncsiak*, azaz csak passzív támadásokat kezdeményeznek, nem hagyva nyomot a támadás során. Attól függően, hogy a szerver becsületes vagy becsületes, de kíváncsi, megkülönböztetünk **becsületes és becsületes, de kíváncsi modelleket**. A [21] és [20] becsületes modelleket használnak. A javasolt protokollban becsületes, de kíváncsi modellt tételezünk fel.

2. Tétel. A javasolt jelszóregisztrációs protokoll ellenáll az online támadásoknak a becsületes, de kíváncsi modellben, feltételezve, hogy a MAC *egzisztenciálisan hamisíthatatlan egy adaptív választott üzenet alapú támadás során*, *Bilineáris Diffie-Hellman* nehéz probléma, továbbá a bilineáris leképezéseket az általános bilineáris csoport modellben, illetve a hash függvényeket véletlen orákulumnak tekintjük.

3. Tétel. A javasolt jelszóregisztrációs protokoll ellenáll az offline támadásoknak a véletlen orákulum modellben, ha a bilineáris leképezés egyirányú leképezés és a kliens gyengén korrupt.

Összehasonlítva a hatékonyságot más regisztrációs protokollokkal (1. táblazat) az eredmény azt mutatja, hogy az általunk javasolt regisztrációs protokoll hatékonyabb a többi javaslathoz képest.

Sémák	Kliens	Szerver	Teljes
BPR- 2 szerveres	1,4 s	0,68 s	2,76 s
BPR - VPAKE	0,72 s	0,67 s	1,5 s
TLS			0,168 s
Mi javaslatunk	0,072 s	0,023s	0,095 s

1. táblázat. A protokollok végrehajtási ideje (másodpercben)

Skálázható, jelszó és küszöbszámon alapuló hitelesítés okos otthonokhoz

A 6. fejezetben bemutatunk egy küszöbszámon és jelszón alapuló, elosztott, kölcsönösen hitelesített kulcsmegegyezés és kulcskonfirmáció protokollt egy okos otthon környezetben. A javasolt felhőalapú hitelesítési sémánkban ([14]) feltételezzük, hogy a felhőkiszolgálók mindig elérhetőek. Az okos otthoni rendszerekben azonban az eszközök különféle típusúak lehetnek, ami azt jelenti, hogy egyes eszközök akkumulátorról működnek, míg mások korlátozott erőforrásokkal rendelkeznek, és előfordulhat, hogy nem elérhetőek a felhasználó számára. Figyelembe véve az okos otthonok ezen tulajdonságát egy új titokmegosztási technikával működő felhasználói hitelesítési sémát javaslunk, ahol megköveteljük, hogy a dinamikusan választható n darab készülék közül k legyen elérhető, ahol $k \leq n$. A fejezet eredményeit a Huszti Andreával és Kovács Szabolccsal közös cikkünk [15] tartalmazza.

A protokoll tervezése során fontos a megfelelő jelszóhasználat beállítása és a végpontok közötti biztonságos kommunikáció elérése. A javasolt protokoll egy méretezhető és robusztus séma, ahol a sikeres szótártámadáshoz $k-1$ darab okos otthoni eszközt (k a jelszó küszöbszám) kell kompromittálnia a támadónak. A tudományos irodalomban Bagherzandi a [1] dolgozatban egy jelszóval védett titokmegosztási (PPSS) és küszöbszámon alapuló megoldást mutat be. Jarecki a [18] publikációban javasol egy egykörös op-

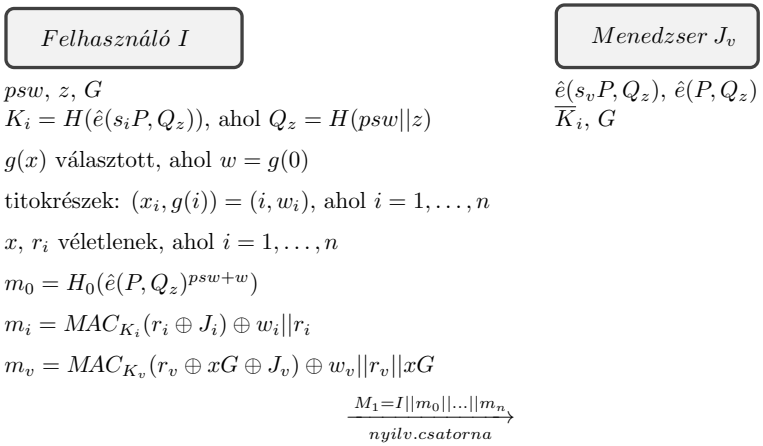
timális PPSS-sémát, amely mindösszesen két üzenetküldést tartalmaz. Ezek a megoldások azonban nem skálázhatóak. Işler és Küpçü a [16, 17] publikációiban hasonló szempontokat vesznek figyelembe (skálázhatóság, robusztusság, jelszóhasználat stb.), viszont jobban alkalmazhatóak felhő környezetben, és protokolljaik tartalmaznak tárolószolgáltatókat. A mi megoldásunk okos otthon környezetre lett kialakítva, ahol $n \geq 10$ eszköz és $o \geq 5$ küszöbszám esetén (o a hitelesítéshez szükséges IoT eszközök száma) jobb hatékonysági eredményt érünk el.

Két résztvevője van a protokollunknak. Az egyik az *IoT rendszer*, amely tartalmazza a eszközkezelőt és az IoT-eszközöket $(J_1, \ldots, J_n)$. A másik résztvevő a *felhasználó* (I), amely kéri a szolgáltatásokat és az adatokat. A protokollban titokmegosztást alkalmazunk, ahol (k, n) küszöbszám sémát használunk. Egy titkos S egész felosztható n részre oly módon, hogy $k \leq n$ lesz a titokrészek küszöbszáma, amellyel ki kell tudjuk számítani az S egészt. Így $k - 1$ vagy annál kevesebb titokrésszel nem lehet meghatározni az S egészt. A jelszó létrehozásához Shamir-féle titokmegosztást alkalmazunk az IoT-eszközökön.

A beállítási fázis során a felhasználó kiválaszt egy psw jelszót, majd a kliens szoftver generál és biztonságosan tárol egy véletlenszerű z salt értéket és egy véletlenszerű polinomot a psw Shamir-féle titokmegosztásához. Legeneráljuk az s_i titokrészeket, ahol $i = 1, \ldots, n$ és az eszközök elküldik és tárolják az $\hat{e}(P, Q_z)$ és $\hat{e}(s_i P, Q_z)$ értékeket, ahol $\hat{e}(,)$ a bilineáris leképezést jelöli, $Q_z = H(psw||z)$ és P a $\mathbb{G}$ egy generátora, ahol $\mathbb{G}$ egy q-adrendű additív csoport, ahol q egy nagy prím. A hitelesítési fázisban a kliensszoftver kiszámolja a jelszómegosztáson alapuló, hosszú élettartamú szimmetrikus titkos kulcsokat $K_i = H(\hat{e}(s_i P, Q_z))$. Ha a felhasználó új eszközöket akar beállítani az okos otthon rendszerbe, akkor meg kell adnia a jelszót a kliensszoftvernek, amely új extra s_i megosztásokat generál ugyanarra a polinomra, ahol $i > n$. Így a konstrukció tartalmazza a skálázhatóság tulajdonságát. Legyen E egy véges $\mathbb{F}$ test felett definiált elliptikus görbe, $G \in E(\mathbb{F})$ pedig egy generátorelem. Minden IoT-eszköz rendelkezik egy szimmetrikus titkosítási kulccsal

$(\overline{K}_1, \ldots, \overline{K}_n)$, amely a menedzsereszköznek küldött üzenetek bizalmasságát és hitelesítését biztosítja.

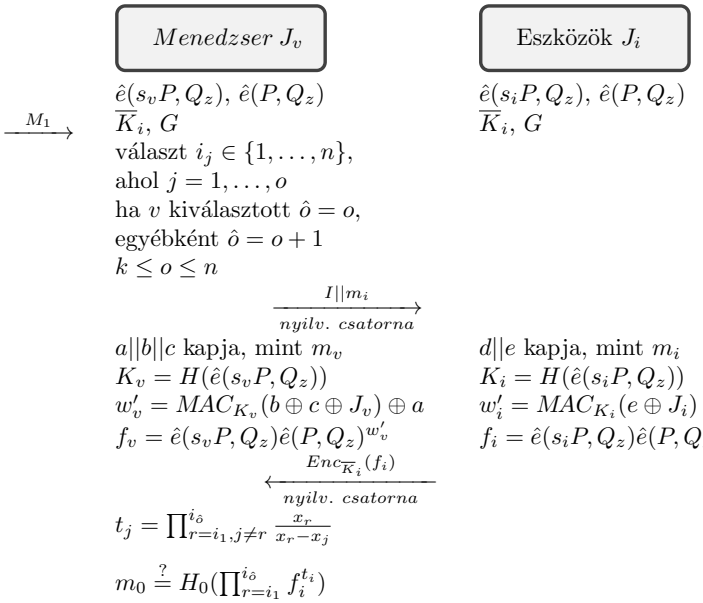
A hitelesítési szakasz három fő szakaszból áll. Az első fázist (10. ábra) a kliens szoftver hajtja végre. A rendszer egy titkos, véletlenszerű w hitelesítési értéket választ, és a Shamir-féle titokmegosztással felosztja. Ezek a titokrészek, a w, a jelszó és a salt-on alapuló m_0 hash érték biztonságosan átkerül az eszközkezelőhöz.



10. ábra. Hitelesítés - Kliens folyamat

A hitelesítés második fázisában (11. ábra) a véletlenszerűen kiválasztott okos otthoni eszközök kiszámolják jelszó titokrészeiken alapuló hosszú élettartamú szimmetrikus titkos kulcsukat $K_i = H(\hat{e}(s_iP, Q_z))$, összeállítják és ellenőrzik az $\hat{e}(P, Q_z)^{w+psw}$ értéket, amely a jelszó, a salt és a w titkos, véletlenszerű hitelesítési értéken alapszik.

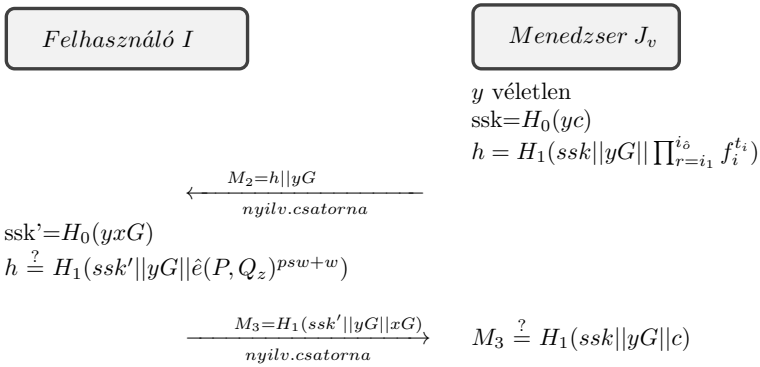
A harmadik fázisban (12. ábra) egy titkos szimmetrikus kulcsot cserél a felhasználó és az eszközkezelő, majd a felhasználó ellenőrzi, hogy az okos otthon rendszere képes-e kiszámítani az $\hat{e}(P, Q_z)^{w+psw}$ értéket, tehát az eszközök rendelkeznek-e a megfelelő jelszó titkokkal



11. ábra. Hitelesítés - Eszközök folyamat

és salt-tal.

Részletes biztonsági elemzést nyújtunk a javasolt AKC protokollról. Az egyik alapvető biztonsági követelmény a résztvevők kölcsönös hitelesítése, amely megakadályozza, hogy a támadók érvényes felhasználónak vagy eszközkezelőnek adják ki magukat, és illegálisan hozzáférhessenek az érzékeny adatokhoz. Egy másik biztonsági cél a generált kulcs titkossága, azaz a támadónak nem szabad semmilyen információval rendelkeznie az új munkamenetkulcsról. Protokollfuttatás során egy véletlenszerűen kiválasztott új munkamenetkulcsot kell kicserélni a résztvevők között, és fontos, hogy a protokoll végrehajtását ne lehessen sikeresen befejezni egy korábban kicserélt kulccsal. A feleknek képesnek kell lenniük ellenőrizni, hogy



12. ábra. Hitelesítés - Végső folyamat

a másik fél ismeri-e és képes-e használni az új munkamenetkulcsot.

Figyelembe vesszük az ismert kulcs biztonságot és forward secrecy tulajdonságokat is. Az ismert kulcs biztonság lényege, hogy megőrzi a munkamenetkulcsok biztonságát abban az esetben is, ha egy munkamenetkulcsot felfedtek. Tehát egy munkamenetkulcs nyilvánosságra hozatala nem veszélyeztetheti más munkamenetkulcsok biztonságát. A forward secrecy tulajdonság fennáll, ha egy vagy több entitás hosszú távú kulcsai sérülnek és ez nincs hatással a korábbi munkamenetkulcsok titkosságára. A felhasználó szerepét vagyis a felhasználó lépéseit a protokollban AVISPA eszközzel formalizáltuk. Alkalmaztuk az OFMC és a CL-AtSe modellt, és végrehajtottuk a támadószimulációt. A biztonsági elemzés az mutatja, hogy kölcsönös hitelesítés megsértésére, illetve a munkamenetkulcs titkosságának sérülésére nem talált támadást az AVISPA.

A hatékonysági elemzéshez kiválasztottunk egy küszöbszám hitelesítési rendszert [17], amely leginkább hasonló a mi rendszerünkhöz. A futási időket összehasonlítottuk a két rendszernél különböző számú eszköz és küszöbszám esetén. A [11] szerint 2022-ben háztartásonként átlagosan 500 IoT eszköz lesz csatlakoztatva, ezért

az eszközök nagy száma és a küszöbszámok figyelembe vétele kiemelt szempont. Az általunk javasolt rendszer jobb eredményt ad $n \geq 10$ számú eszköz és $o \geq 5$ küszöbszám esetén (2. táblazat).

Küszöbszám	**2-5**	**3-6**	**5-10**
Işler, Küpçü - DSPP	0,00806	0,01171	0,01833
Javasolt megoldás	0,0150602	0,0150648	0,0150766

2. táblázat. Teljesítmény összehasonlítás (másodpercben).

Manapság a számítási kapacitás optimalizálása és a megfelelő biztonság fontos szempont az IoT eszközöknél. A gyártási költség befolyásolja ezen eszközök képességeit, azonban gondoskodnunk kell a biztonságról. Ezeket a szempontokat is figyelembe vettük a protokollunk kialakítása során.

Irodalomjegyzék

[1] A. Bagherzandi, S. Jarecki, N. Saxena, Y. Lu, *Password-protected secret sharing.* In: ACM Conference on Computer and Communications Security (2011), pp. 433–444.

[2] M. Bellare, D. Pointcheval, P. Rogaway, *Authenticated key exchange secure against dictionary attacks.*, In Bart Preneel, editor, Advances in Cryptology - EUROCRYPT2000, volume 1807 of Lecture Notes in Computer Science, Springer, (2000), pp. 139–155.

[3] S. M. Bellovin, M. Merritt, *Encrypted key exchange: Password-based protocols secure against dictionary attacks.*, In Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on. IEEE, (1992), pp. 72–84.

[4] C. Bertok, A. Huszti, S. Kovacs, N. Olah, *Provably Secure Identity-Based Remote Password Registration.*, Cryptology ePrint Archive, (2022). pp.

[5] S. Blake-Wilson, D. Johnson, A. Menezes, *Key agreement protocols and their security analysis*, Proceedings of the sixth IMA International Conference on Cryptography and Coding, LNCS 1355, (1997), pp. 30–45.

[6] X. Boyen, *Hidden credential retrieval from a reusable password.*, In: Proceedings of the 4th International Symposium on Information, ACM, (2009), pp. 228–238.

[7] V. Boyko, P. MacKenzie, S. Patel, *Provably secure password-authenticated key exchange using diffie-hellman*, In EUROCRYPT'00, volume 1807 of LNCS, Springer, (2000), pp. 156–171.

[8] N. Chen, R. Jiang, *Security Analysis and Improvement of User Authentication Framework for Cloud Computing*, Journal of Networks, **9(1)**, (2014), pp. 198–203.

[9] A. J. Choudhury, P. Kumar, M. Sain, *A Strong User Authentication Framework for Cloud Computing*, Proceedings of IEEE Asia -Pacific Services Computing Conference,(2011), pp. 110–115.

[10] W. Ford, B.S. Kaliski, *Server-Assisted Generation of a Strong Secret from a Password*, Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (2000), pp. 176–180.

[11] T. Gu, P. Mohapatra, *Bf-iot: Securing the iot networks via fingerprinting-based device authentication,* In: 2018 IEEE 15Th international conference on mobile ad hoc and sensor systems (MASS). IEEE, (2018), pp. 254–262.

[12] A. Huszti, N. Olah, *A simple authentication scheme for clouds*, Proceedings of IEEE Conference on Communications and Network Security (CNS), (2016), pp. 565–569.

[13] A. Huszti, N. Olah, *Security analysis of a cloud authentication protocol using applied pi calculus.*, International Journal of Internet Protocol Technology, 12(1), (2019), pp. 16–25.

[14] A. Huszti, N. Olah, *Provably Secure Scalable Distributed Authentication for Clouds.*, International Conference on Cryptology and Network Security,Springer, Cham, (2020), pp. 188–210.

[15] A. Huszti, Sz. Kovács, N. Olah, *Scalable, password-based and threshold authentication for smart homes.*, Int. J. Inf. Secur. 21, https://doi.org/10.1007/s10207-022-00578-7, (2022), pp. 707— 723.

[16] D. Işler, A. Küpçü, *Threshold single password authentication.*, ESORICS Data Privacy Management, Cryptocurrencies and Blockchain Technology, (2017), pp. 143–162.

[17] D. Işler, A. Küpçü, *Distributed Single Password Protocol Framework.*, IACR Cryptol. ePrint Arch., 976., (2018),

[18] S. Jarecki, A. Kiayias, H. Krawczyk, *Round-optimal password-protected secret sharing and T-PAKE in the password-only model*, In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg. (2014), pp. 233–253.

[19] J. Katz, R. Ostrovsky, M. Yung, *Efficient password-authenticated key exchange using human-memorable passwords.*, In EUROCRYPT 2001. Springer, (2001), pp. 475–494.

[20] F. Kiefer, M. Manulis, *Blind password registration for verifier-based PAKE.*, In: Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography. (2016), pp. 39–48.

[21] F. Kiefer, M. Manulis, *Blind password registration for two-server password authenticated key exchange and secret sharing protocols.*, In: International Conference on Information Security. Springer, Cham, (2016), pp. 95–114.

[22] P. MacKenzie, T. Shrimpton, M. Jakobsson, *Threshold password-authenticated key exchange*, In CRYPTO 2002. Springer, (2002), pp. 385–400.

[23] M. D. Raimondo, R. Gennaro, *Provably secure threshold password-authenticated key exchange*, International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, (2003), pp. 507–523.

[24] M. E. S. Saeed, Q. Y. Liu, G. Y. Tian, B. Gao, F. Li, *AKAI-oTs: authenticated key agreement for Internet of Things.* Wireless Netw 25, (2019), pp. 3081–3101.

[25] T. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, C. Chen, *An authenticated key exchange protocol for multi-server architecture in 5G networks.* IEEE Access 8, https://doi.org/10.1109/ACCESS.2020.2969986, (2020), pp. 28096–28108.

Előadások

1. Securing cloud authentication, *International Conference on Applied Informatics*, Eger, Hungary, 2017.

2. DECAP-Distributed Extensible Cloud Authentication Protocol, *Cryptacus: Workshop & MC meeting*, Nijmegen, Netherlands 2017.

3. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *Central European Conference on Cryptology* , Smolenice, Slovakia, 2018.

4. Security Analysis of Identity-based Password Registration for Distributed Systems, *OGIK 2019*, Budapest, Hungary, 2019.

5. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *Central European Conference on Cryptology*, Telc, Czech Republic, 2019.

6. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *International Conference for Internet Technology and Secured Transaction (ICITST-2019)* , London, United Kingdom, 2019.

7. Identity-based Password Registration for Clouds, *The 11th International Conference on Applied Informatics*, Eger, Hungary, 2020.

8. Provably Secure Scalable Distributed Authentication for Clouds, *19th International Conference on Cryptology and Network Security*, Online, 2020.

9. Biztonságos autentikáció okos otthonokra *ADA 2020* Online, 2020.

10. Provably secure authentication for smart homes, The 1st Conference on Information Technology and Data Science Debrecen, Hungary, 2021.

11. Scalable, password-based and threshold authentication for Smart Homes, *Central European Conference on Cryptology* , Online, 2021.

12. Secure Blind Password Registration, *Central European Conference on Cryptology* , Smolenice, Slovakia, 2022.

DEBRECENI EGYETEM
EGYETEMI ÉS NEMZETI KÖNYVTÁR
H-4002 Debrecen, Egyetem tér 1, Pf.: 400
Tel.: 52/410-443, e-mail: publikaciok@lib.unideb.hu

Nyilvántartási szám: DEENK/412/2022.PL
Tárgy: PhD Publikációs Lista

Jelölt: Oláh Norbert
Doktori Iskola: Informatikai Tudományok Doktori Iskola
MTMT azonosító: 10067778

A PhD értekezés alapjául szolgáló közlemények

Idegen nyelvű tudományos közlemények hazai folyóiratban (1)

1. Bertók, C., Huszti, A., Kovács, S. Z., **Oláh, N.**: Provably Secure Identity-Based Remote Password Registration.
   Publ. Math. Debr. "Accepted by Publisher" (-), 1-33, 2022. ISSN: 0033-3883.
   IF: 0.698 (2021)

Idegen nyelvű tudományos közlemények külföldi folyóiratban (2)

2. Huszti, A., Kovács, S., **Oláh, N.**: Scalable, password-based and threshold authentication for smart homes.
   Int. J. Inf. Secur. 21, 707-723, 2022. ISSN: 1615-5262.
   DOI: http://dx.doi.org/10.1007/s10207-022-00578-7
   IF: 2.427 (2021)

3. Huszti, A., **Oláh, N.**: Security analysis of a cloud authentication protocol using applied pi-calculus.
   Int. J. Internet Prot. Technol. 12 (1), 16-25, 2019. ISSN: 1743-8209.
   DOI: http://dx.doi.org/10.1504/IJIPT.2019.10019901

Idegen nyelvű konferencia közlemények (4)

4. Huszti, A., **Oláh, N.**: Provably Secure Scalable Distributed Authentication for Clouds.
   In: Cryptology and Network Security. Eds.: Stephan Krenn, Haya Shulman, Serge Vaudenay, Springer, Cham, 188-210, 2020, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 12579) ISBN: 9783030654108

5. Huszti, A., **Oláh, N.**: Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems.
   In: International Conference for Internet Technology and Secured Transactions, Infonomics Society, London, 69-75, 2019. ISBN: 9781913572068

6. Huszti, A., **Oláh, N.**: Identity-Based Cloud Authentication Protocol.
   In: The 11th Conference of PhD Students in Computer Science : Volume of short papers, University of Szeged, Szeged, 33-36, 2018.

DEBRECENI EGYETEM
EGYETEMI ÉS NEMZETI KÖNYVTÁR
H-4002 Debrecen, Egyetem tér 1, Pf.: 400
Tel.: 52/410-443, e-mail: publikaciok@lib.unideb.hu

7. Huszti, A., **Oláh, N.**: A simple authentication scheme for clouds.
In: 2016 IEEE Conference on Communications and Network Security. Ed.: Jie Wu, IEEE Computer Society, Washingon, 565-569, 2016. ISBN: 9781509030651

További közlemények

Idegen nyelvű konferencia közlemények (1)

8. Huszti, A., Kovács, S. Z., **Oláh, N.**: Hybrid anonymous message broadcast for VANETs.
In: 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, Piscataway, 103-108, 2021. ISBN: 9781665428545

A közlő folyóiratok összesített impakt faktora: 3,125
A közlő folyóiratok összesített impakt faktora (az értekezés alapjául szolgáló közleményekre): 3,125

A DEENK a Jelölt által az iDEa Tudóstérbe feltöltött adatok bibliográfiai és tudománymetriai ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján elvégezte.

Debrecen, 2022.09.01.

DEBRECENI EGYETEM * Egyetemi és Nemzeti Könyvtár *

Short thesis for the degree of doctor of philosophy (PhD)

Designing Secure Authentication Schemes for Distributed Systems

by Norbert Oláh

Supervisor: Dr. Andrea Pintér-Huszti

University of Debrecen
Doctoral School of Informatics

Debrecen, 2022

Summary

One of the essential issues during online communication is the secure authentication between the participants. The proper authentication serves to avoid the different attacks (e.g. impersonation attack). However, in the case of improper authentication, user access control, confidentiality and integrity of user data are not provided. The authentication schemes require several security requirements, which depend on the attributes of environments. One of the most widely used authentication methods is based on short secrets like passwords. The registration process must be executed before the authentication, but it receives insufficient attention in the scientific literature.

The present dissertation demonstrates three new entity authentication schemes and a user registration protocol, which is necessary before the first identity verification. Distributed identity verification is carried out by multiple participants to secure cloud computing services and smart home environments. Via formal analysis we demonstrate that the protocols fulfil the necessary security requirements. Our solutions are more efficient than the current practical and theoretical schemes.

The first chapter contains the scientific background of the user authentication schemes and solutions.

In the second chapter, we detail the cryptographic primitives applied in our protocols and the necessary preliminaries.

Chapter 3 covers automated security analysis tools and gives the

details of the concept of provable security.

In Chapter 4, two distributed authentication protocols are proposed for cloud services.

In Chapter 5, a password registration scheme is demonstrated based on the identity-based cryptography, *i.e.* both the user and the service provider are authenticated by their short-lived identity-based secret key.

In Chapter 6, we present a threshold and password-based, distributed, mutual authenticated key agreement with key confirmation protocol for a smart home environment.

Cloud Authentication Protocol Using a Merkle Tree

In Chapter 4.1, a two-factor authentication scheme for cloud computing services using a Merkle tree is demonstrated ([12]). In contrast to [8, 9] and the practical solutions, where only *one* cloud server verifies the users' authenticity, our solution applies multiple servers for user authentication. We have extended the scheme in [13] and also provided a security analysis in applied pi calculus. In our protocol, an attack can be successful only if the adversary possesses all password shares known by the servers. Comparing the efficiency of our authentication phase to the work of [8, 9], our scheme is more efficient, since only hash calculations are performed. The results of this chapter are contained in our papers ([12, 13]). These papers are joint work with Andrea Huszti.

The user is authenticated with a static and a one-time password on the service provider's side at a randomly selected server that can verify the one-time password by using a Merkle tree (Figure 13). A leaf of the Merkle tree is the hash of a password share, and the root element is verified in order to confirm the correctness of the whole one-time password. The protocol has three phases: registration, authentication, and synchronization.

In the registration phase (Figure 14), the secret keys are exchanged generating a large amount of one-time passwords between the user and the cloud servers. Each cloud server (C_i) possesses an

asymmetric key pair: $SK_{C_i} = (y_i, z_i)$, $PK_{C_i} = (g^{y_i}, g^{z_i})$, where g is a generator element of a cyclic group, and $y_i, z_i \in \mathbb{Z}_q$ are random.

In the authentication phase (Figure 15), the mutual authentication between the user and a randomly chosen cloud server (C_v), fur-

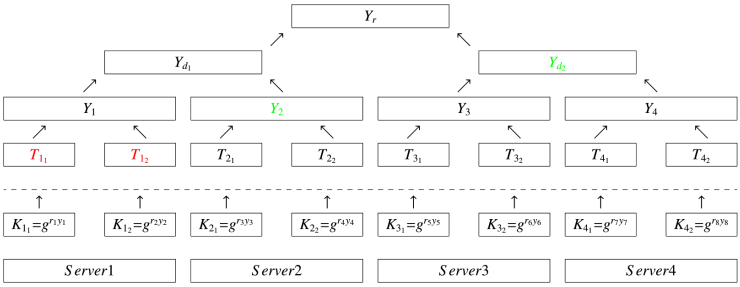


Figure 13. A Merkle tree with 8 leaves

U	C_i	AS
ID, PW, X salt (r_i, s_i) secret (g^{r_i}, g^{s_i}) $K_{i_1}=g^{r_i y_i}$, $K_{i_2}=g^{s_i z_i}$ $T_i=(T_{i_1},T_{i_2})=(H(K_{i_1}), H(K_{i_2}))$ building the Merkle tree	$(y_i,\ z_i)$ secret key $(g^{y_i},\ g^{z_i})$ public key	
$\xrightarrow{ID,\ (g^{r_i},g^{s_i}),\ H(PW\|\|X)}$	$K_i=(g^{r_i y_i},\ g^{s_i z_i})$ $Y_i=H(H(K_{i_1})\|\|H(K_{i_2}))$	
	$\xrightarrow{Y_i}$	Y_r building the Merkle tree
	$\xleftarrow{Y_r}$	
	$< ID, K_i, H(PW\|\|X), Y_r >$	

Figure 14. Registration

thermore a MAC key exchange are processed. A message authentication key (MAC) exchange is also provided to guarantee data origin integrity for the latter interactive communication. After authentication, a synchronization step (Figure 16) follows and the password of the selected server is updated with the path associated with the tree.

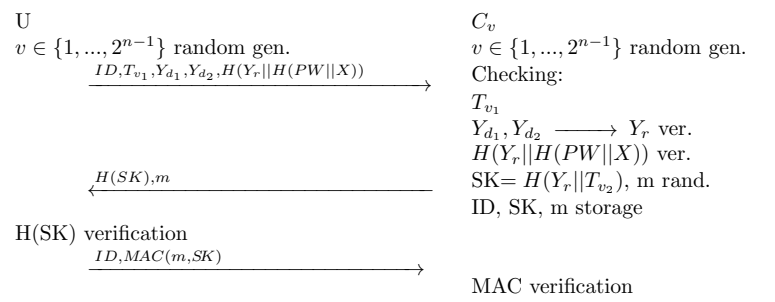


Figure 15. Authentication

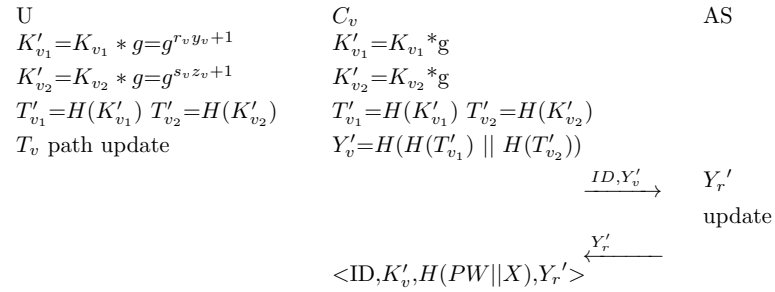


Figure 16. Synchronization

For security analysis, the protocol is formatted in ProVerif and the results of the ProVerif test show that the specified security criteria are met. We analyse this protocol as a key exchange scheme

hence the typical security requirements for mutual entity authentication schemes, and also the key related requirements are considered. We formalise the protocol in ProVerif and prove four properties:

1. Authentication of both parties
   (a) Authentication of users: Adversaries must not be able to impersonate a legal user and achieve illegal access to the user data.
   (b) Authentication of the server: Adversaries must not be able to impersonate a legal cloud server.
2. Secrecy of the MAC key: During the key exchange the newly generated key is a confidential datum and an adversary must not have any information about the new key.
3. Key freshness: During a protocol run a new, randomly chosen key must be exchanged so that a protocol execution could not be successfully finished with an old, already used key exchanged.
4. Both parties must verify that the other party knows and is able to use the new MAC key.

We apply injective correspondences for the security analysis of the user and server authentication. All the queries return with the value true, which means that user and server authentications and key secrecy hold in our model and ProVerif do not find an attack. Assuming a successful mutual authentication, *key freshness and key confirmation* hold, as well.

Scalable Distributed Authentication for Cloud Services

In Chapter 4.2, we propose a multi-server password-based authenticated key exchange scheme (Figures 17 - 19). In contrast to

other threshold password-based protocols applying secret-sharing algorithms ([1, 3, 6, 7, 23, 18, 17, 22, 19]), even if we share the password information among the servers, it is not reconstructed from the shares to verify it. To show that the proposed protocol is provably secure, we introduce the threshold hybrid corruption model. Unlike [6, 10] we provide a detailed security analysis based on the Bellare and Rogaway model. Compared to other schemes, we also consider the scalability property, which is one of the main requirements for clouds. We demonstrate a new way of generating a strong secret (e.g. long-lived key) from a password, which is also suitable for scalability. In the IoT environment, an authenticated key exchange (AKE) protocol is presented ([24]) on wireless sensor networks. They focus on the key shares and establish the authenticated key between Wireless sensor networks and the cloud server, which performs a centralized authentication. Another variant of AKE is demonstrated in [25] which includes a permanent Control Server and cloud servers on 5G network. Our solution differs from these papers ([25, 24]) since we can scale the generated long-lived keys on the user's and the provider's sides as well. Compared to our earlier proposed protocol ([13]), we use secret splitting technique and we also achieve the scalable property. The results of this chapter are contained in our paper ([14]). This paper is a joint work with Andrea Huszti.

Our protocol results in a session key, which provides the confidentiality of the subsequent messages between the participants. The protocol has two phases. During registration, the client sets password-based long-lived keys with all the n servers. We propose a simple solution in which the client accesses the long-lived keys by using a password. We assume that a client software is running on the client device (e.g. smartcard, mobile phone etc.) that requires a password from the user to initiate the authentication process. After the client gives the password, the client software generates the long-lived keys and the execution of authentication begins. The correctness of the password is verified by the servers and not by the client software, hence a client device does not store any information

about the password. The client randomly chooses k servers out of n servers for authentication. During authentication, a server is only able to calculate the challenge value w given by the client with the knowledge of the symmetric, long-lived key K_i ($i \in \{1, \ldots, k\}$) which is generated from the client password. *KKDF* denotes a Keyed Key Derivation Function that for a message m and a *key* generates a secret key K. The authentication server (J_v) authenticates the client by verifying the correctness of all the k challenge values received from the participating servers.

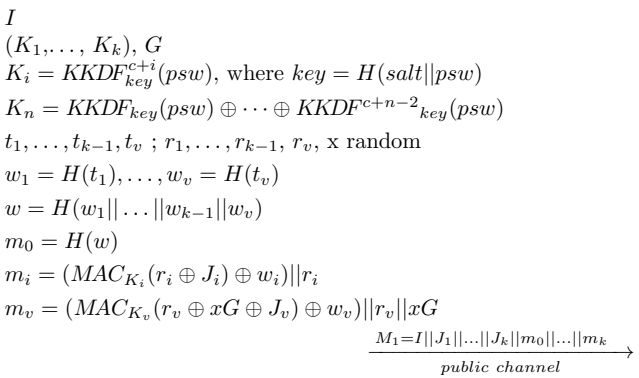


Figure 17. Authentication - Client process

In our proposed protocol, servers communicate on secure channels. We prefer one randomly chosen server that communicates with the client, hence the client does not need to communicate with all the k servers in parallel and build secure channels. During the design of the protocol, the efficiency of authentication is ensured by MAC and other fast cryptographic algorithms (hash, xor operation, symmetric encryption). The protocol is provably secure and the necessary adversary model and the formal proof are given. We assume that $\mathcal{A}$ is allowed to make the `Send, Reveal, Corrupt, Test` queries.

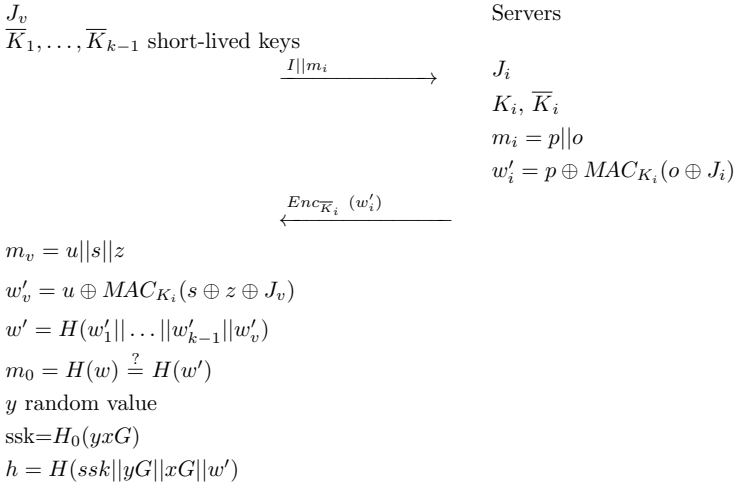


Figure 18. Authentication - Cloud servers communication

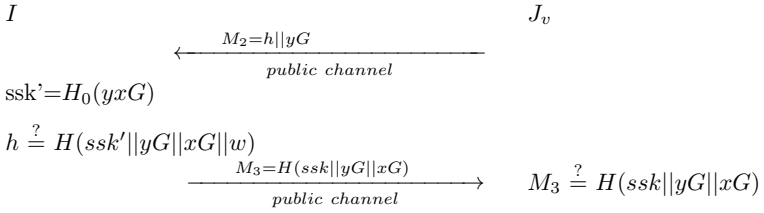


Figure 19. Authentication - Final process

We apply distributed authentication, thus we extend the model with the concept of threshold hybrid corruption. We assume that the participants can be corrupt in our proposition. A model is a *strong corruption model* ([2]) if long-lived keys $K_{I,J}$ and all the values stored by the participant I (e.g. randomly chosen secret values) are transferred to $\mathcal{A}$ during the protocol run. In the case of the *weak*

corruption model, only the long-lived keys $K_{I,J}$ are transferred or replaced, the adversary does not completely compromise the machine. Other values generated and stored during the protocol run are not revealed. We introduce a new threshold hybrid corruption model.

1 Definition. We call a model *threshold hybrid corruption model* if the client is uncorrupted, there are at least $n - k + 1$ uncorrupted servers out of the n servers and k servers are chosen randomly for authenticated key exchange with key confirmation protocol (AKC). Moreover, the server chosen to communicate with the client is

1. uncorrupted, or
2. corrupted weakly and among the remaining servers there is at least one uncorrupted.

In order to give the definition of a secure AKC protocol, we need to review the definitions of conversation and matching conversation from [5].

Matching conversation formalizes real-time communication between entities I and J, it is also necessary to be specified for the authentication property of an AKC protocol. We give the definition of the event $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ that is a modified version of the definition given in [5]. We leave out the requirement that $J \in Server$ is uncorrupted. In our multi-server setting, each client communicates with a server that can be weakly corrupted if there is at least one uncorrupted server from the k servers.

2 Definition. $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ denotes an event when in a protocol P in the presence of an adversary $\mathcal{A}$ assuming a threshold hybrid corruption model, there exists

1. a client oracle $\prod_{I,J}^{s}$ which is accepted, but there is no server oracle $\prod_{J,I}^{t}$ having a matching conversation with $\prod_{I,J}^{s}$, or
2. a server oracle $\prod_{I,J}^{s}$ which is uncorrupted and accepted, but there is no client oracle $\prod_{J,I}^{t}$ having a matching conversation with $\prod_{I,J}^{s}$, or

3. a server oracle $\prod_{I,J}^{s}$ which is weakly corrupted and accepted, but there is no client or no uncorrupted server oracle having a matching conversation with $\prod_{I,J}^{s}$.

In order to give the definition of a secure AKC, it is essential to define the notion of *freshness* and redefine the *benign adversary*.

3 Definition. A $k+1$-tuple of oracles containing one client and k server oracles is fresh if in the threshold hybrid corruption model the client oracle and the server oracle with which it has had a matching conversation are unopened. We call an oracle *fresh* if it is an element of a fresh $k+1$-tuple.

4 Definition. An adversary is called *benign* if it is deterministic, and restricts its action to choosing a $k+1$ tuple of oracles containing one client and k server oracles, and then faithfully conveying each flow from one oracle to the other, with the client oracle beginning first.

5 Definition. We introduce that a protocol is a *secure AKC protocol* if

1. In the presence of the benign adversary the client oracle and the server oracle communicating with the client oracle always accept holding the same session key ssk, and this key is distributed uniformly at random on $\{0,1\}^{\kappa}$.

and if for every adversary $\mathcal{A}$

2. If in a threshold hybrid corruption model there is a server oracle $\prod_{I,J}^{l}$ having matching conversations with a client oracle and if $\prod_{I,J}^{l}$ is weakly corrupted, $\prod_{I,J}^{l}$ has matching conversation with an uncorrupted server oracle, then the client oracle and oracle $\prod_{I,J}^{l}$ both accept and hold the same session key ssk.

3. The probability of $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ is negligible.

4. If the tested oracle is fresh, then $Adv^{\mathcal{A}}(\kappa)$ is negligible.

1 Theorem. *The proposed protocol is a secure AKC protocol in the random oracle model, assuming MAC is* universally unforgeable under an adaptive chosen-message attack *and symmetric encryption scheme is* indistinguishable under chosen plaintext attack*, moreover,* ECCDH assumption holds *in the elliptic curve group.*

Efficiency is an important aspect during the design of our protocol. In the protocol, the session key is generated by ECDH key exchange, and the other operations are hash and xor operations, which are extremely fast.

Provably Secure Identity-Based Remote Password Registration

In Chapter 5, we demonstrate the Certificate-Less Secure Blind Registration Protocol (CLS-BPR) on the Identity-Based Cryptography, *i.e.* both the user and the service provider are authenticated by their short-lived identity-based secret key. The proposed protocol suits our smart home user authentication scheme where the values of the bilinear map are stored on the IoT devices. Our cloud scheme ([13]) can also be easily modified with the proper long-lived key setting to be compatible with our registration scheme.

For secure storage of the password, a bilinear map with a salt is applied, therefore in case of an offline attack the adversary is forced to calculate a computationally expensive bilinear map for each password candidate and salt, which slows down the attack. In contrast to traditional registration schemes, our solution does not require a Transport Layer Security (TLS) channel and can also omit the associated certificate management, which can be efficiently implemented in a corporate or educational institution. According to our implementation, our protocol is more cost-effective than the TLS-based and the other blind solutions ([20, 21]). It is not necessary to manage certificates or execute costly zero knowledge (ZK) proof. Unlike other schemes ([20, 21]) besides the password hashing scheme we also consider the interactions, when the password information is

sent securely. Consequently, we prove that our solution is secure against online attacks as well. We introduce the definition of a secure password registration scheme, provide an adversarial model and show that our scheme is provably secure. Our registration is flexible, which is optimal for Single Sign-On (SSO) and Kerberos, but it is also suitable for systems where different passwords must be applied for each service. The bilinear map of the password and the salt can be used as a long-lived symmetric key and applied for entity authentication or session key generation. The results of this chapter are contained in our paper ([4]). This paper is a joint work with Andrea Huszti, Csanád Bertók and Szabolcs Kovács.

The protocol consists of a Setup and a Registration phase (Figure 20, Figure 21). During the Setup, system parameters and keys are generated for the participants. Let P be a generator of $\mathbb{G}$, where $\mathbb{G}$ additive group of order q for some large prime q. Choose a random $\alpha \in \mathbb{Z}_q^*$ and generate parameters $P, \alpha P$. The master secret key for the system is α. Identities denoted by ID_C and ID_S and public keys are derived, *i.e.* $PK_C = Q_C = tr(ID_C)$ and $PK_S = Q_S = tr(ID_S)$. Since our password hashing scheme uses bilinear pairings ($\hat{e}$) on elliptic curves, we need an efficient way to map passwords first into $\mathbb{Z}_p$, where p is a large prime, then these points of $\mathbb{Z}_p$ into a point on the curve. Let denote tr this function. The Private Key Generator calculates the participants' secret keys $SK_C = \alpha Q_C$ and $SK_S = \alpha Q_S$. In the Registration phase, the clients send their password information to the server and confirm that the server has received the verification value. The protocol meets all the necessary requirements, including password secrecy, mutual authentication and resistance against offline attacks.

We provide a security model that considers resistance against online attacks in addition to offline attacks. Our proposed model take the whole registration process into account unlike [21] and [20]. We have regard to all communication messages between the client and the server as well. Hence mutual authentication of the participants and password secrecy are also studied during transmission. Ad-

Client (C)	PKG	Server (S)
	$\alpha \in \mathbb{Z}_q^*$ (msk)	$x \in \mathbb{Z}_q^*$ secret key
	Public information: $P, \alpha P, x\alpha P$	
$Q_C = tr(ID_C)$ (PK_C)		$Q_S = tr(ID_S)(PK_S)$
αQ_C (SK_C)		αQ_S (SK_S)

Figure 20. Setup

versary $\mathcal{A}$ is allowed to make the queries that model adversarial attacks. These queries are `Send, Corrupt, Reveal, Test, Execute` and `Finalise`.

We define the security goals for password registration protocols and consider the whole registration process assuming the minimum requirements. We introduce the definition of secure registration:

6 Definition. A protocol is a *secure registration protocol* if

- Online resistance:

Client (C)		Server (S)
$z \in \mathbb{Z}_q^*$ random, psw password		
$R = tr(psw)$		
$m = \hat{e}(Q_S, zx\alpha P + \alpha Q_C) \cdot \hat{e}(zP, R)$		
$K = H(\hat{e}(zP, R))$		
$V = H(\hat{e}(Q_S, zx\alpha P + \alpha Q_C)\|\|K)$		
	$\xrightarrow{Q_C, zP, m, V}$	
		$x \cdot zP$
		$\overline{K} = m \cdot \hat{e}(\alpha Q_S, xzP + Q_C)$
		$K' = H(\overline{K})$
		$V \stackrel{?}{=} H(\hat{e}(\alpha Q_S, xzP + Q_C)\|\|K')$
		$r \in \mathbb{Z}_q^*$ random
		$MAC_{K'}(r)$
	$\xleftarrow{Q_S, MAC_{K'}(r), r}$	
$MAC_{K'}(r) \stackrel{?}{=} MAC_K(r)$		
		Store: $Q_C, \hat{e}(zP, R), zP$

Figure 21. Password registration protocol

1. In the presence of the *benign adversary* the client oracle and the server oracle communicating with the client oracle are always accepted. The server stores the password verification value confirmed by the client.

and for every adversary $\mathcal{A}$

2. If there is an uncorrupted client oracle having matching conversations with an uncorrupted server oracle, then they are always accepted. The server stores the password verification value confirmed by the client;
3. For uncorrupted server and client oracles the probability of $\texttt{No-Matching}^{\mathcal{A}}(\kappa)$ is negligible;
4. For the tested oracle $Adv^{\mathcal{A}}(\kappa)$ is negligible. If it is a client oracle, then it is unopened;

- Offline resistance:

5. If for all dictionaries D_n adversary $\mathcal{A}$ generates at most t tuples (C, S, psw), then

$$Pr[\texttt{Finalise}(C, S, psw) = 1] \leq \frac{t}{2^{\beta_{D_n}} \cdot t_{pre}} + \mu(\kappa),$$

where $\mu(\kappa)$ is negligible, $\frac{t}{2^{\beta_{D_n}} \cdot t_{pre}}$ denotes the probability that $\mathcal{A}$ finds psw by trying t number of (C, S, psw) tuples, β_{D_n} is the min-entropy for dictionary D_n and t_{pre} denotes the computational cost to calculate the input value of the one-way function from the password.

Protocol security is considered in the random-oracle model, the hash functions and the bilinear map are supposed as random oracles. We define two security models. In the case of client-server protocols, clients usually are assumed to be malicious, *i.e.* they deviate form the steps of the protocol, they apply any type of strategy to attack. The servers providing some service are usually considered to be honest, meaning they do not launch any attack or honest-but-curious, *i.e.* they initiate only passive attacks, not leaving any trace of the

attack. Depending on whether the server is honest or honest-but-curious, we differentiate **honest and honest-but-curious models.** In [21] and [20] honest models are used.

2 Theorem. *The proposed password registration protocol is resistant against online attacks in the honest-but-curious model, assuming MAC is* existentially unforgeable under an adaptive chosen-message attack*, solving the* Bilinear Diffie-Hellman problem *is computationally infeasible, moreover, the bilinear map is considered in the generic bilinear group model and the hash functions are random oracles.*

3 Theorem. *The proposed password registration protocol is resistant against offline attacks in the random oracle model if the bilinear map is a one-way pairing and the client is weakly corrupted.*

Our registration protocol achieves better results in term of efficiency and Table 3 demonstrates this comparison.

Scheme	**Client**	**Server**	**Full**
BPR- two server	1,4 s	0,68 s	2,76 s
BPR - VPAKE	0,72 s	0,67 s	1,5 s
TLS			0,168 s
Our proposition	0,072 s	0,023s	0,095 s

Table 3. The execution time of the protocols

Scalable, Password-Based and Threshold Authentication for Smart Homes

In Chapter 6, a threshold and password-based, distributed, mutual authenticated key agreement with key confirmation protocol for a smart home environment is presented. In our proposed cloud authentication scheme ([14]), we assume that the cloud servers are always available. However, the devices can be of various types in smart home systems, which means some devices are battery-powered

or resource-constrained and might not be available. We need to consider this property of smart homes, and we propose a new smart home user authentication scheme with a secret sharing technique, where we require k devices to be available out of n ones, which can be chosen dynamically. The results of this chapter are contained in our paper ([15]). This paper is a joint work with Andrea Huszti and Szabolcs Kovács.

The protocol is designed to achieve the password-only setting, and end-to-end security if the chosen IoT devices are also authenticated besides the user. The proposed protocol is a scalable and robust scheme, which forces the adversary to corrupt $k-1$ smart home devices, where k is the threshold, in order to perform an offline dictionary attack. In the scientific literature, a threshold Password-Protected Secret Sharing (PPSS) scheme was formalized by Bagherzandi et. al. ([1]). Jarecki et. al. ([18]) present the first round-optimal PPSS scheme, requiring just one message from user to server and from server to user, and prove its security in the challenging password-only setting where users do not have access to an authenticated public key. However, it is not scalable. These recommendations ([17, 16]) considered similar properties to our proposition (scalability, robustness, password usage, etc.). However, these are more suitable in the cloud environment and their protocols contain storage providers. Our solution is recommended typically for a smart home environment and provided a better result for $n \geq 10$ number of devices and for $o \geq 5$ thresholds.

There are two participants in our protocol. One of them is the *IoT system* including the manager device and the IoT devices $(J_1, \ldots, J_n)$ and the other one is the *user* (I), who queries services and data. We apply secret sharing where we use a (k, n) threshold scheme. A secret S can be divided into n shares in a way that $k \leq n$ will be the threshold number of the shares which we need to be able to compute S. Thus $k-1$ or fewer shares leave S completely undetermined. We apply Shamir's secret sharing threshold scheme for the IoT devices to construct the password.

During the setup phase, the user chooses a password psw, the

client software generates and securely stores a random salt value z and a random polynomial for the Shamir secret sharing of psw. The secret shares s_i, where $i = 1, \ldots, n$ are generated and values $\hat{e}(P, Q_z)$ and $\hat{e}(s_i P, Q_z)$ are sent and stored by the devices, where $\hat{e}(,)$ denotes the bilinear map and $Q_z = H(psw||z)$. The password share-based long-lived symmetric secret keys $K_i = H(\hat{e}(s_i P, Q_z))$ are calculated for the authentication during the authentication phase. If a user wants to set new devices to the smart home system, they need to give the password to the client software, which generates new extra shares s_i for the same polynomial, where $i > n$. This way the construction includes the property of scalability. Let E denotes an elliptic curve defined over a finite field $\mathbb{F}$ and $G \in E(\mathbb{F})$ be a generator element. Each IoT devices possess symmetric encryption keys $(\overline{K}_1, \ldots, \overline{K}_n)$ for authenticated encryption of the messages sent to the manager device.

The authentication phase consists of three main subphases. The first subphase (Figure 22) is carried out by the client software. A secret, random authentication value w is chosen and split into shares with Shamir secret sharing. These shares and a hash value m_0 based on the authentication value w, the password and the salt value are transferred securely to the manager device. During the second subphase (Figure 23), randomly chosen smart home devices calculate their password share-based long-lived symmetric secret keys $K_i = H(\hat{e}(s_i P, Q_z))$, construct and also verify the user's knowledge of the value $\hat{e}(P, Q_z)^{w+psw}$, which is based on the password, the salt and the secret, random authentication value w.

In the third subphase (Figure 24), a secret symmetric key is exchanged between the user and the manager device and the user checks whether the smart home system is able to calculate $\hat{e}(P, Q_z)^{w+psw}$, *i.e.,* whether the devices possess the password shares and the salt.

A detailed security analysis of the proposed AKC protocol is provided. One of the indispensable security requirements is the mutual authentication of the participants. The secure mutual authentication of participants prevents adversaries from impersonating a legal

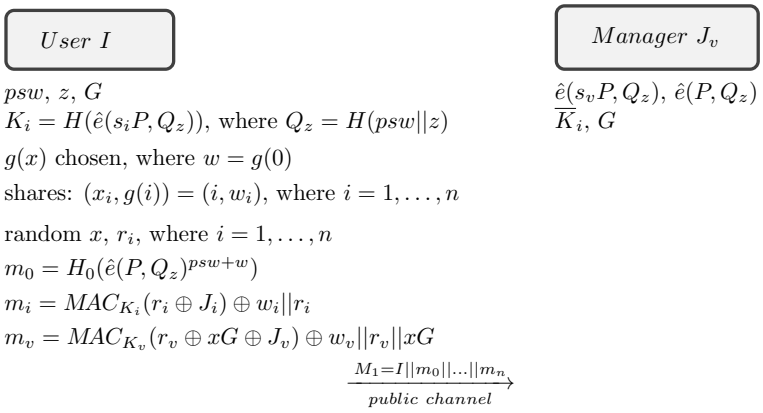


Figure 22. Authentication - Client process

user or the device manager and gaining illegal access to sensitive data. Another security goal is key secrecy, *i.e.,* an adversary must not possess any information about the new key. During a protocol run, a new randomly chosen session key should be exchanged between the participants, a protocol execution cannot be successfully completed with an old key exchanged before. At the end, parties should be able to verify that the other party knows and is able to use the new session key. Known-key security and forward secrecy properties are also considered. Known-key security preserves the security of other session keys after disclosure of a session key. Disclosure of a session key should not jeopardize the security of other session keys. Forward secrecy holds if the long-term secrets of one or more entities are compromised but the secrecy of previous session keys is not affected. The user's role including the user's steps in the protocol was formalized in AVISPA. We apply the OFMC and CL-AtSe and executed the attacker simulation. The results of the security analysis show that the attacker is not able to impersonate the legal participants or obtain the session key. We have selected a

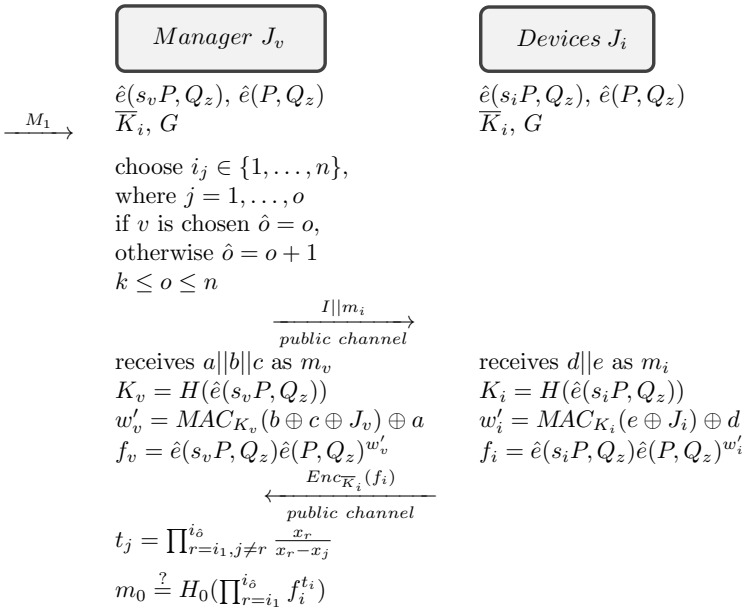


Figure 23. Authentication - Devices' process

threshold authentication system [17], which is similar to our system. We compare their runtime results with ours for the different number of devices and thresholds. According to [11], on average 500 devices will be connected per household in 2022, hence a large number of devices and thresholds should be considered. Our proposition provides a better result for $n \geq 10$ number of devices and for $o \geq 5$ threshold (Table 4).

Today, achieving computing capacity and adequate security are important considerations for IoT devices. The cost of manufacturing affects the capabilities of these devices, however, we need to ensure security. These aspects are also taken into account during the design of our protocol.

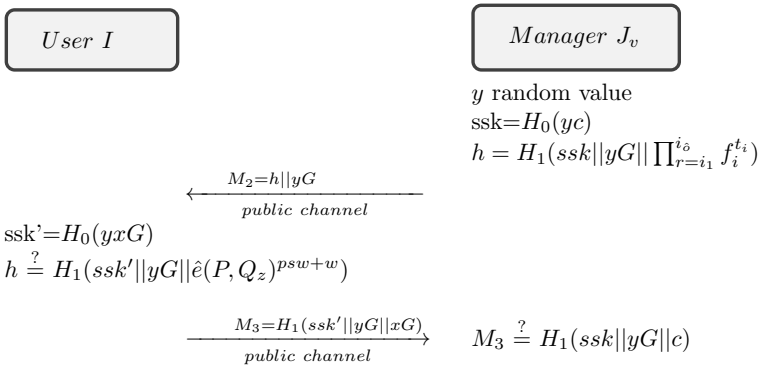


Figure 24. Authentication - Final process

Threshold	**2-5**	**3-6**	**5-10**
Işler, Küpçü - DSPP	0,00806	0,01171	0,01833
Our proposition	0,0150602	0,0150648	0,0150766

Table 4. Performance comparison (in seconds).

Bibliography

[1] A. Bagherzandi, S. Jarecki, N. Saxena, Y. Lu, *Password-protected secret sharing.* In: ACM Conference on Computer and Communications Security (2011), pp. 433–444.

[2] M. Bellare, D. Pointcheval, P. Rogaway, *Authenticated key exchange secure against dictionary attacks.*, In Bart Preneel, editor, Advances in Cryptology - EUROCRYPT2000, volume 1807 of Lecture Notes in Computer Science, Springer, (2000), pp. 139–155.

[3] S. M. Bellovin, M. Merritt, *Encrypted key exchange: Password-based protocols secure against dictionary attacks.*, In Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on. IEEE, (1992), pp. 72–84.

[4] C. Bertok, A. Huszti, S. Kovacs, N. Olah, *Provably Secure Identity-Based Remote Password Registration.*, Cryptology ePrint Archive, (2022). pp.

[5] S. Blake-Wilson, D. Johnson, A. Menezes, *Key agreement protocols and their security analysis*, Proceedings of the sixth IMA International Conference on Cryptography and Coding, LNCS 1355, (1997), pp. 30–45.

[6] X. Boyen, *Hidden credential retrieval from a reusable password.*, In: Proceedings of the 4th International Symposium on Information, ACM, (2009), pp. 228–238.

[7] V. Boyko, P. MacKenzie, S. Patel, *Provably secure password-authenticated key exchange using diffie-hellman*, In EUROCRYPT'00, volume 1807 of LNCS, Springer, (2000), pp. 156–171.

[8] N. Chen, R. Jiang, *Security Analysis and Improvement of User Authentication Framework for Cloud Computing*, Journal of Networks, **9(1)**, (2014), pp. 198–203.

[9] A. J. Choudhury, P. Kumar, M. Sain, *A Strong User Authentication Framework for Cloud Computing*, Proceedings of IEEE Asia -Pacific Services Computing Conference,(2011), pp. 110–115.

[10] W. Ford, B.S. Kaliski, *Server-Assisted Generation of a Strong Secret from a Password*, Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (2000), pp. 176–180.

[11] T. Gu, P. Mohapatra, *Bf-iot: Securing the iot networks via fingerprinting-based device authentication,* In: 2018 IEEE 15Th international conference on mobile ad hoc and sensor systems (MASS). IEEE, (2018), pp. 254–262.

[12] A. Huszti, N. Olah, *A simple authentication scheme for clouds*, Proceedings of IEEE Conference on Communications and Network Security (CNS), (2016), pp. 565–569.

[13] A. Huszti, N. Olah, *Security analysis of a cloud authentication protocol using applied pi calculus.*, International Journal of Internet Protocol Technology, 12(1), (2019), pp. 16–25.

[14] A. Huszti, N. Olah, *Provably Secure Scalable Distributed Authentication for Clouds.*, International Conference on Cryptology and Network Security,Springer, Cham, (2020), pp. 188–210.

[15] A. Huszti, Sz. Kovács, N. Olah, *Scalable, password-based and threshold authentication for smart homes.*, Int. J. Inf. Secur. 21, https://doi.org/10.1007/s10207-022-00578-7, (2022), pp. 707— 723.

[16] D. Işler, A. Küpçü, *Threshold single password authentication.*, ESORICS Data Privacy Management, Cryptocurrencies and Blockchain Technology, (2017), pp. 143–162.

[17] D. Işler, A. Küpçü, *Distributed Single Password Protocol Framework.*, IACR Cryptol. ePrint Arch., 976., (2018),

[18] S. Jarecki, A. Kiayias, H. Krawczyk, *Round-optimal password-protected secret sharing and T-PAKE in the password-only model*, In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg. (2014), pp. 233–253.

[19] J. Katz, R. Ostrovsky, M. Yung, *Efficient password-authenticated key exchange using human-memorable passwords.*, In EUROCRYPT 2001. Springer, (2001), pp. 475–494.

[20] F. Kiefer, M. Manulis, *Blind password registration for verifier-based PAKE.*, In: Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography. (2016), pp. 39–48.

[21] F. Kiefer, M. Manulis, *Blind password registration for two-server password authenticated key exchange and secret sharing protocols.*, In: International Conference on Information Security. Springer, Cham, (2016), pp. 95–114.

[22] P. MacKenzie, T. Shrimpton, M. Jakobsson, *Threshold password-authenticated key exchange*, In CRYPTO 2002. Springer, (2002), pp. 385–400.

[23] M. D. Raimondo, R. Gennaro, *Provably secure threshold password-authenticated key exchange*, International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, (2003), pp. 507–523.

[24] M. E. S. Saeed, Q. Y. Liu, G. Y. Tian, B. Gao, F. Li, *AKAIoTs: authenticated key agreement for Internet of Things.* Wireless Netw 25, (2019), pp. 3081–3101.

[25] T. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, C. Chen, *An authenticated key exchange protocol for multi-server architecture in 5G networks.* IEEE Access 8, https://doi.org/10.1109/ACCESS.2020.2969986, (2020), pp. 28096–28108.

List of talks

1. Securing cloud authentication, *International Conference on Applied Informatics,* Eger, Hungary, 2017.

2. DECAP-Distributed Extensible Cloud Authentication Protocol, *Cryptacus: Workshop & MC meeting,* Nijmegen, Netherlands 2017.

3. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *Central European Conference on Cryptology* , Smolenice, Slovakia, 2018.

4. Security Analysis of Identity-based Password Registration for Distributed Systems, *OGIK 2019,* Budapest, Hungary, 2019.

5. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *Central European Conference on Cryptology,* Telc, Czech Republic, 2019.

6. Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems, *International Conference for Internet Technology and Secured Transaction (ICITST-2019)* , London, United Kingdom, 2019.

7. Identity-based Password Registration for Clouds, *The 11th International Conference on Applied Informatics,* Eger, Hungary, 2020.

8. Provably Secure Scalable Distributed Authentication for Clouds, *19th International Conference on Cryptology and Network Security*, Online, 2020.

9. Biztonságos autentikáció okos otthonokra *ADA 2020* Online, 2020.

10. Provably secure authentication for smart homes, The 1st Conference on Information Technology and Data Science Debrecen, Hungary, 2021.

11. Scalable, password-based and threshold authentication for Smart Homes, *Central European Conference on Cryptology* , Online, 2021.

12. Secure Blind Password Registration, *Central European Conference on Cryptology* , Smolenice, Slovakia, 2022.

UNIVERSITY of DEBRECEN

UNIVERSITY AND NATIONAL LIBRARY
UNIVERSITY OF DEBRECEN
H-4002 Egyetem tér 1, Debrecen
Phone: +3652/410-443, email: publikaciok@lib.unideb.hu

Registry number: DEENK/412/2022.PL
Subject: PhD Publication List

Candidate: Norbert Oláh
Doctoral School: Doctoral School of Informatics
MTMT ID: 10067778

List of publications related to the dissertation

Foreign language scientific articles in Hungarian journals (1)

1. Bertók, C., Huszti, A., Kovács, S. Z., **Oláh, N.**: Provably Secure Identity-Based Remote Password Registration.
   Publ. Math. Debr. "Accepted by Publisher" (-), 1-33, 2022. ISSN: 0033-3883.
   IF: 0.698 (2021)

Foreign language scientific articles in international journals (2)

2. Huszti, A., Kovács, S., **Oláh, N.**: Scalable, password-based and threshold authentication for smart homes.
   Int. J. Inf. Secur. 21, 707-723, 2022. ISSN: 1615-5262.
   DOI: http://dx.doi.org/10.1007/s10207-022-00578-7
   IF: 2.427 (2021)

3. Huszti, A., **Oláh, N.**: Security analysis of a cloud authentication protocol using applied pi-calculus.
   Int. J. Internet Prot. Technol. 12 (1), 16-25, 2019. ISSN: 1743-8209.
   DOI: http://dx.doi.org/10.1504/IJIPT.2019.10019901

Foreign language conference proceedings (4)

4. Huszti, A., **Oláh, N.**: Provably Secure Scalable Distributed Authentication for Clouds.
   In: Cryptology and Network Security. Eds.: Stephan Krenn, Haya Shulman, Serge Vaudenay, Springer, Cham, 188-210, 2020, (Lecture Notes in Computer Science, ISSN 0302-9743 ; 12579) ISBN: 9783030654108

5. Huszti, A., **Oláh, N.**: Provably Secure Authenticated Key Agreement with Key Confirmation for Distributed Systems.
   In: International Conference for Internet Technology and Secured Transactions, Infonomics Society, London, 69-75, 2019. ISBN: 9781913572068

6. Huszti, A., **Oláh, N.**: Identity-Based Cloud Authentication Protocol.
   In: The 11th Conference of PhD Students in Computer Science : Volume of short papers, University of Szeged, Szeged, 33-36, 2018.

UNIVERSITY of DEBRECEN

UNIVERSITY AND NATIONAL LIBRARY
UNIVERSITY OF DEBRECEN
H-4002 Egyetem tér 1, Debrecen
Phone: +3652/410-443, email: publikaciok@lib.unideb.hu

7. Huszti, A., **Oláh, N.**: A simple authentication scheme for clouds.
In: 2016 IEEE Conference on Communications and Network Security. Ed.: Jie Wu, IEEE Computer Society, Washingon, 565-569, 2016. ISBN: 9781509030651

List of other publications

Foreign language conference proceedings (1)

8. Huszti, A., Kovács, S. Z., **Oláh, N.**: Hybrid anonymous message broadcast for VANETs.
In: 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, Piscataway, 103-108, 2021. ISBN: 9781665428545

Total IF of journals (all publications): 3,125
Total IF of journals (publications related to the dissertation): 3,125

The Candidate's publication data submitted to the iDEa Tudóstér have been validated by DEENK on the basis of the Journal Citation Report (Impact Factor) database.

01 September, 2022