# Felhasználó-azonosítás gépelési szokások alapján

X. Erdélyi Tudományos Diákköri Konferencia Kolozsvár, 2007. május 26-27

*Szerző:*

Bócsi Botond Attila

Babes-Bolyai Tudományegyetem
Matematika és Informatika Kar
Informatika szak, III. év

*Témavezető:*

Dr. Csató Lehel

Adjunktus
Babes-Bolyai Tudományegyetem
Matematika és Informatika Kar
Programozási nyelvek és módszerek tanszék

# 1. Bevezetés

A XXI. század embere számára a számítógép elengedhetetlen. A mindennapok kényelme szinte elképzelhetetlen nélküle, az információszerzés új korszakában járunk: a hagyományos levelezést felváltja az elektronikus levelezés (email), a telefonálást a chatelés, a újságolvasást a digitális média böngészése, sőt az emberek hajlandóak ennél jóval fontosabb dolgaikat is a számítógépre bízni: pénzüket, személyes adataikat. Fejlődőben van az interneten való vásárlás, az online jegyrendelés. Ez az előretörés nem megállítható, sőt még csak azt sem lehet mondani, hogy szabályozható, el kell fogadni, hogy nemsokára a számítógépektől fogunk függeni (feltéve, hogy ez még nem következett be). Az emberek sokszor felelőtlenül bánnak bankszámlaszámukkal, kódjaikkal, jelszavaikkal és hajlandóak elhinni, hogy a pénz, amit küldenek, oda megy, ahova ők ezt szánták.

Az érem másik oldalán találhatóak a kiszolgálók, akiknek az a céljuk, hogy minél nagyobb megbízhatósággal azonosítsák ügyfeleiket, alkalmazottaikat. Fontos, hogy a biztonság szintjét a védeni kívánt adatok fontossága függvényében válasszuk meg. Ez azt jelenti, hogy egy egyszerű felhasználónak, aki internetezésre, játszásra, szövegszetkesztésre használja számítógépét nem érdemes komolyabb lépéseket tennie a biztonságért (az operációs rendszer által felajánlott jelszó használata elegendő), mert a védeni kívánt adatok *értéke* kicsi. Egy célhardverre való beruházás lehet, hogy többe kerülne, mint a potenciális adatlopásból származó kár. Azzal is növelni lehetne a biztonságot, ha időnként megkérdeznénk a jelszót, ezzel azt biztosítva, hogy a jogos személy ül a gépnél, de így elviselhetetélenül kényelmetlenné válna a rendszer használata és ez is fontos szempont, szem előtt kell tartani.

Az igazán komoly helyeke, mint például a bankokban nem létezik egy általános rendszergazda, minden feladathoz külön rendszergazda tartozik. Van olyan személy, akit azért alkalmaznak, hogy reggel bejelentkezzen a központi szerverre és este megszakítsa a kapcsolatot vele, van, aki a biztonsági másolatokért felel, és még sok más. Léteznie kell egy olyan adminisztrátornak is, aki ezen feladatok kiosztását, karbantartását végzi. Ennek a jelszavát senki sem ismeri, ez szét van osztva egy többtagú bizottság között, és csak akkor lehet használni, ha mindenki jelen van, begépeli a

jelszó hozzá tartozó részét (hexa karakterek sorozata) és figyeli az elvégzett módosításokat [13]. És még ilyen megszorítások mellett is előfordul, hogy illetéktelen kezek jutnak bizalmas adatokhoz.

Látszik, hogy a mai számítástechnikában nélkülözhetetlen a felhasználók azonosítása, módszert kell találni arra, hogy kellően kicsi hibaszázalékkal meg tudjuk határozni, hogy egy illető az-e akinek kiadja magát. Erre három lehetőség adódik:

- tudás alapján
- birtokolt tárgy alapján
- tulajdonság alapján

Mindegyik módszernek megvannak a maga előnyei, hátrányai, de az nyílvánvaló, hogy ezeknek az egyidejű alkalmazása számottevően megnöveli a rendszer biztonságát, hiszen az előbb említett megoldások függetlenek egymástól, így ezek hibaszázalékai összeszorzódnak. Például, ha az egyik azonosítás 80%-os biztonságú, egy másik 90%-os, akkor ezek egyidejű használata 98%-os megbízhatóságot eredményez.

A **tudás alapján** való azonosítás azon alapszik, hogy a felhasználónak valamilyen ismeretét kéri számon. Ez a típus a legelterjedtebb, kézenfekvő példa rá a jelszavak használata, amikor mondjuk a regisztrációnál megadott szót kell újra begépelni. Egy másik példa a PIN kódok használata a bankkártyák, illetve smartcardok esetén. A legnagyobb hátránya az, hogy az igényelt tudást elfelejthetjük vagy felelőtlenül felírjuk valahova és így ez mások számára is elérhetővé válik.

Habár az előbbi mód a legismertebb, nem ezt használták először. A **birtok alapján** történő azonosításra példát már nagyon régről találhatunk, sőt napjainkban is igen gyakran használjuk, mondjuk amikor bedugjuk a kulcsot az ajtóba vagy az autóba, esetleg amikor a rendőrnek a személyi igazolványunkat mutatjuk. Egy kifinomultabb példa a bankkártya vagy a smartcard. Hátránya az, hogy a birtololt tárgyat el lehet veszíteni, ellophatják és annak típusától függően könnyebbennehezebben másolhatják.

A bankkártya és a smartcard szerepelt mind a két példában, ez azért van, mert ezeknél birtok alapú **és** tudás alapú azonosításra is szükség van.

A harmadik módszerről a következő fejezetben lesz szó.

# 2. Biometria

A **tulajdonság alapú** azonsítás (más néven biometriai azonosítás) napjainkban kezd teret hódítani magának, bár nem újkeletű. Itt olyan tulajdonságokat vizsgálunk, amik az illető személyre jellemzőek, ezek lehetnek fizikai jellemzők vagy szerzett tulajdonságok, személyiségjegyek. A felismerés nem valamilyen közvetett úton történik (tudás, tulajdon által), hanem magát a személyt probáljuk felismertetni a számítógéppel.

Az első biometriai azonosítás az aláírás használata volt a XVII. század végén, ez azóta is használatban van és megfelelően biztonságosnak mondható. Az következő utalás erre a módszerre 1858-ban történt [15], amikor *Sir William Herschel* ujjlenyomatát használta hitelesítésre. Azóta az ujjlenyomat a rendőrség bevált eszköze a bűnözők azonosítására. A számítógépek megjelenésével új lehetőségek nyíltak meg a biometriai azonosítás továbbfejlesztésére, kibővítésére. Ilyen új ágak például az arcfelismerés, hangfelismerés, retintavizsgálat, a kéz vagy fül geometriájának a vizsgálata és a végő megoldás a DNS-vizsgálat lenne.

A biometriai azonosítás legkiemelkedőbb előnye az, hogy az azonosításra használt *eszközt* nem lehet ellopni, legalábbis nagyon körülményes (például valakinek az újjait vagy a szemét), illetve másolása is nehézkes és nemegyszer rendkívül drága. Ezek után feltevődik a kérdés, hogy akkor miért nem ezt használjuk. Ennek az elsődleges oka az, hogy a különböző biometriai jellemzők mérése nagyon drága, ezért csak a nagyon nagy biztonságot igénylő rendszerek engedhetik meg maguknak. Egy másik hátránya az, hogy néhány biometriai jellemző időben változik. Ilyen jellemző például az ember arca, hangja (esetleg keze, ujjlenyomata egy baleset következtében). Ezekre a változásokra sokszor nagyon nehéz *megtanítani* a felismerő rendszert.

A számítógépes klaviatúra elterjedésével lehetővé vált az azonosítás a gépelési szokások alapján. Ez egyértelműen biometriai jellegű, mivel eléggé sok gépelés után a folyamat automatikussá válik, így személyiségre jellemző tulajdonságokat tartal-

maz. Használatával a biometriai azonosítás előnyeit élvezhetjük és kiküszöbölhetjük néhány hátrányát. Annak bizonyítása, hogy minden személy egyedi gépelési ritmussal rendelkezik nem egyszerű, csak statisztikai megközelítés lehetséges, de számottevő eredmény ezzel kapcsolatban még nem született. A gépelés által való azonosításnak az a legnagyobb előnye, a többi biometrikus módszerhez képest, hogy lényegesen olcsóbb, hiszen a mérőeszköz egy átlagos billentyűzet, így bármilyen munkaállomáson elérhető a használata. Sajnos a tulajdonságok időben való változásának hátrányát itt sem tudjuk kiküszöbölni.

Az ötletet az adta, hogy a II. Világháború idején az üzeneteket morze kódok formájában továbbító katonák ismerték egymás gépelési ritmusát, így meg tudták mondani, hogy hiteles-e a kapott üzenet. Az első erre irányuló kutatást az Egyesült Államok tudományos kutatási ügynöksége (U.S. National Science Foundation) készítette 1980-ban [16].

# 3. Felhasználó azonosítása

Lássuk, hogy milyen tulajdonságokat lehet vizsgálni, amikor valakit a gépelése alapján szeretnénk felismerni. A legkézenfekvőbb mérendő jellemvonás az, hogy mennyi ideig tartja lenyomva a felhasználó az egyes billentyűket, tehát az az idő, ami a billentyű lenyomásától a billentyű felengedéséig telik el (*hold time*). Egy másik triviális sajátosság, hogy mennyi idő telik el két billentyűhasználat között, vagyis az az idő, ami egy billentyű felengedése és a következő billentyű lenyomása között telik el (*latency time*). Érdesek megjegyezni, hogy ez a második érték lehet negatív is, hiszen egy nagy tapasztalattal rendelkező gépelő szinte mindig előbb nyomja le a következő billentyűt, minthogy az előbbit felengedné. Az előbb említett jellegzetességek szolgáltatják a legtöbb információt a felhasználóról. Ennek finomításához be lehet vezetni azt, hogy az előbbi időket minden betűpár (*digraph*) esetén megvizsgáljuk, így pontosabb és részletesebb képet kaphatunk a felhasználóról. Ez persze azzal jár, hogy lényegesen több adatot kell tárolnunk. Ennek a továbbgondolása az, hogy néhány betűhármast (*trigraph*) is vizsgáljunk.

Természetesen más adatokat is mérhetünk, például, hogy egy adott szöveg be-

gépelésének mennyi az összideje, vagy a nagybetűk írásának módját (*capslock*ot használ-e a felhasználó vagy jobb/bal *shift*et). Szintén érdekes lehet a törlések figyelése, néhányan a *backspace*t használják a hibák javítására, mások a vissza billentyűt és *delete*. A törléssel szorosan össze lehet kapcsolni a hibák megjelenésének, ezeknek típusának és gyakoriságának vizsgálatát. Látszik, hogy a probléma nagyon összetett, számos tulajdonság vizsgálatát teszi lehetővé.

Mint minden (nem csak boimetriai) azonosításnak, ennek is valahogyan mérnünk kell a hatékonyságát. Erre két mérőszán létezik. Az egyik az, hogy milyen százalékban utasítunk vissza egy érvényes felhasználót, ennek a megnevezése **FRR** (*False Reject Rate*), a másik az, hogy milyen százalékban fogadunk el egy illetéktelen felhasználót, ennek a megnevezése **FAR** (*False Accept Rate*). A statisztikában ezeket a típusú hibákat első-, illetve másodrendű hibának vagy kockázatnak szokták nevezni.

A gépelési szokások vizsgálat problémájának megfogalmazása többféleképpen is feltevődhet. A legkézenfekvőbb az lenne, ha egy előre rögzített szöveget kellene begépelni. Ezt először Joyce és Gupta [17] használta, amikor a felhasználót keresztneve, családneve, felhasználóneve és jelszava alapján próbálta meg felismerni. Ez rövid szöveg és így eléggé kevés információt hordoz. Egy 2006-os probálkozás [10] hosszabb szövegek alapján (300-600 karakter) próbálta azonosítani a felhasználót, nem is rossz eredménnyel. A végső megoldás az lenne, ha tetszőleges szövegre is működne a felismerés, akár olyankor is, amikor a felhasználó mindennapi tevékenységét végzi. Természetesen erre is voltak kísérletek [8, 10], de ez egy lényegsen nehezebb feladat.

Az első ilyen típusú felismerésre Gaines [16] *T-Teszt*-et alkalmazott, nem sok sikerrel, aztán újabb és újabb megközelítések láttak napvilágot.

## 3.1. Euklideszi távolság

Az első megoldásban csak az egyes billentyűk lenttartásának idejét és/vagy az billentyűpárok használata között eltelt időt mérjük, így az ezek által alkotott vektort kell vizsgálni. Legyen a regisztráláskor készített vektor $R$, ami $N$ elemet tartalmaz, legyenek ezek $r_i$, $i < N$. Hasonló jelöléssel legyen $U$ a belépéskor regisztrált értékeket

tartalmazó vektor. Az ötlet az, hogy tekintsük a két vektort egy-egy pontnak egy $N$ dimenziós térben és számoljunk euklideszi távolságot köztük.

$$D(R,U) = \left[\sum_{i=1}^{N}(r_i - u_i)\right]^{1/2} \tag{1}$$

Minél kisebb ez a $D$ távolság, annál jobban hasonlít a két ember írása.

## 3.2. Nem súlyozott mérték

Egy statisztikai megközelítésben ne csak azt regisztráljuk, hogy mennyi volt az átlaga a mért időknek, hanem azt is, hogy ez milyen szórással történt, vagyis a vizsgálandó vektor minden eleme tartalmazza a következő számnégyest $\mu_i, \sigma_i, o_i, X_i$, ahol $\mu_i$ a regisztráláskor mért értékek mintabeli várható értéke, a $\sigma_i$ a regisztráláskor mért értékek mintabeli szórása, $o_i$ az $i$-edik karakter előfordulásának száma, $X_i$ pedig az azonosításnál mért érték. Ekkor az $X$ valószínűségi változót standardizáljuk, vagyis kivonjuk belőle a $\mu$-t és elosztjuk a $\sigma$-val. Ezen értékek összegzése után egy standart normál eloszlású valószínűségi változót kapunk, ha függetlennek tekintjük őket. Ez a $D$ érték minél közelebb van a nullához, a két gépelés között annál nagyobb a hasonlóság.

$$D(R,U) = \sum_{i=1}^{N}(S_{u_i}) \tag{2}$$

ahol

$$S_{u_i} = \frac{1}{o_{u_i}}\left[\sum_{j=1}^{o_{u_i}} P\left(\frac{X_{ij}^{(u)} - \mu_{r_i}}{\sigma_{r_i}}\right)\right] \tag{3}$$

A módszer azon a feltételezésen alapszik, hogy a gépeléskor regisztrált idők normál eloszlásúak, ezt több szerző is feltételezi [2, 3]. Én is vizsgáltam a normalitását (Jarque-Bera és Lilliefors tesztekkel 0.05-ös szignifikanciaszint mellett) néhány értéknek: a lenyomva tartástást a *space* és *a* billentyűknek és a billentyűk használata közötti időt az *s* és *z*, illetve a *g* és *y* billentyűk között. A *space* billentyű esetén a tesztek 14.28%-ban adtak pozitív választ (normál eloszlásúnak találták a mintát), míg az *a* esetén ez 4.54%. A billentyűhasználat közötti idő tanulmányozásakor már jobb eredményeket kaptam, a *sz* betőpárnál 70.83% és a *gy*-nél 95.45%. Ebből is látszik, hogy billentyűhasználat közötti idővel nagyobb biztonsággal lehet dolgozni.

## 3.3. Súlyozott mérték

Az előbbi megoldást azzal lehet finomítani, hogy az egyes betűpárokat súlyozhatjuk. Nyílván ezeknek a súlyoknak a meghatározása függ a szöveg nyelvétől, amin a felhasználó gépel, így hiába lehet javítani az előbbi eredményeken, használata egy nemzetközi környezetben nem lehetséges, mert a nyelv váltása nagyon gyakran is előfordulhat. Minden nyelvben vannak olyan betűkombinációk, amiket a felhasználó gyakrabban használ és ezek jellemzőbbek rá (Pélául magyarban a *sz*, *gy*, *ny*, *ty*, stb. vagy angolban a *th*, *er*, *at*, stb.). A 2. képlet annyiban változik, hogy a mindegyik értéket egy $w_i$ súllyal megszorzunk:

$$D(R,U) = \sum_{i=1}^{N} (w_{u_i} S_{u_i}) \tag{4}$$

## 3.4. SVM

A legelső megközelítéshez hasonlóan a **Szupport Vektor Gép** (*SVM - Support Vector Machine*) a kapott mintákat egy-egy pontnak tekinti egy $d$ dimenziós térben. A módszer hátránya (ami a gyakorlatban szinte használhatatlanná teszi) az, hogy a *betanításnál* nem csak az érvényes felhasználóknak kell begépelniük jelszavaikat, hanem még jónéhány olyan embernek, aki használni szeretné a rendszert.

A módszer lényege az, hogy legyen $l$ darab pontunk a $d$ dimenziós térben: $x_i$, ahol $i = 1 \dots l$. Minden ponthoz tartozzon egy $y_i \in \{-1, 1\}$ érték, ami 1, ha az $i$-edik minta a jogos felhasználóé és $-1$, ha valamelyik idegentől származik. Ekkor az lenne a feladat, hogy válasszuk ketté a pontokat egy $\mathbf{w}$ hipersíkkal ($\mathbf{wx + b = 0}$), úgy, hogy az egyik oldalon legyenek azon $i$ pontok, amelyeikre $y_i = -1$ és a másikon azok, amelyekre $y_i = 1$. A $\mathbf{w}$ meghatározásánál az is feltételként szerepel, hogy a távolság a sík mindkét oldalán lévő pontoktól maximális legyen, a hipersíkhoz legközelebb lévő pontokat nevezik *szupport vektoroknak*. Így a felhasználó azonosítása annyiból állna, hogy megnézzük, hogy az általa begépelt minta a hipersík *jó* oldalán helyezkedik-e el. Ha az $\mathbf{R}^d$ halmazt leképezzük egy $\aleph$ halmazra egy $\Phi : \mathbf{R}^d \rightarrow \aleph$ függvénnyel, akkor a felismerés döntése átfogalmazható a következő függvény értékének a meg-

határozására [12, 9]:

$$f(x) = sign\left(\sum_i y_i \alpha_i K(x_i, x) - b\right) \tag{5}$$

ahol $\alpha_i \neq 0$, ha az $i$-edik pont *szupport vektor* és $K(x_i, x_j) = \Phi(x_i)\Phi(x_j)$.

## 3.5. Egyéb módszerek

Egyéb módszereket is kidolgoztak, például Bayes osztályozó segítségével történő felismerést [4], vagy a neurális hálók használatát, de ez utóbbinak elérhető irodalmával nem találkoztam.

Egy 2006-os próbálozásban [10] nem vizsgálták minden billentyűpár esetén az időket, csak a leggyakrabban előforduló betűpároknál, néhány kontrol billentyűnél, a *shift* esetén, figyelték a gépelés összidejét és még az egérkattintásokra is hangsúlyt fektettek. A hasonlóságot euklideszi képlettel számolták és az eredmények eléggé jók lettek.

Léteznek kereskedelmi szoftverek is, ilyen például *http://www.biopassword.com* vagy a *http://www.psylock.com* (és még sok más), de ezek módszerei nem publikusak sőt szerintem a közzétett eredményeket sem tekinthetjük objektívnek.

## 3.6. Saját megközelítés

Kétféle tesztet végeztem: egyet rövid, rogzített jelszavak esetén és egyet hosszabb, tetszőleges szövegekre, figyeltem a billentyűk lenttartásának idejét és külön a használatuk közti eltelt időt. 66 különböző billentyő esetén vizsgáltam az előbb említett tulajdonságokat, ezek a következők: *‘, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, =, backspace, q, w, e, r, t, y, u, i, o, p, [, ], \ , a, s, d, f, g, h, j, k, l, ;, ’, enter, z, x, c, v, b, n, m, ,, ., /, space*, illetve a numpad esetén */, *, -, 7, 8, 9, +, 4, 5, 6, 1, 2, 3, 0, ., enter*. Az gépeléskor regisztrált adatokat nemcsak az aktuális azonosításhoz használtam, hanem le lettek mentve, így később, egy új módszer kipróbálásakor is felhasználhatóak és az erdeményeket össze lehet vetni az eddig elértekkel.

Az első tesztben egy rögzített 5-10 hosszúságú szó háromszori begépelése alapján történt az azonosítás. A regisztrálásnál a *tulajdonos* a jelszavát legalább 25-ször kellett beírja egymás után.

A második tesztben egy tetszőleges magyar szöveg esetén történt a felismerés, ami legalább 150 karakter hosszú volt és a tesztalany szabadon választhatta meg. A regisztrálásnál a *tulajdonos* egy tetszőleges, legalább 2000 karakter hosszú magyar szövegetet kellett begépeljen.

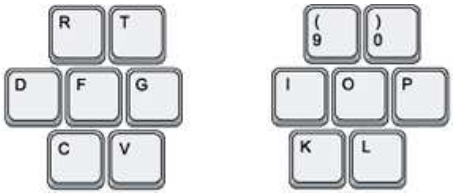
Az első típusú tesztben 9 ember vett részt és összesen 213 bejelentkezés volt, míg az második típusú tesztben 10 ember és összesen 47 bejelentkezés volt.

A saját méréseimet (regisztráció, beléptetés) végző progam C-ben lett írva és Linux alatt működött. Első lépésben a billentyűzetet kellett nyers üzemmódba (*raw mode*) állítani, hogy a billentyűzetről érkező megszakítások (billentyűk lenyomása, felengedése) idejét pontosan lehessen mérni, ezt a **tcsetattr()** és **ioctl()** linuxos rendszerfüggvények segítségével lehet elérni. Ahhoz, hogy a mérések pontosak legyenek a processzor belső időmérését használtam, ehhez a legegyszerübben a **clock_gettime()** függvényen keresztül lehet hozzájutni. Ez elméletileg nanoszekundum pontossággal mér, gyakorlatilag csak microszekundum pontosságú. Számomra a milliszekundum pontosság is elegendő, ezért az adatokat így tároltam, így dolgoztam fel őket.

A Linux alatt való mérés előnye, hogy a program egyszerűen és gyorsan implementálható, az előbb említett függvények segítségével, mivel ezek jól dokumentáltak. Hátránya, hogy kevés ember van, aki tudná használni, egy windows-os verzió megírása az adatok mennyiségének növekedését segíthetné elő.

Mindkét esetben a 3.2 alfejezetben bemutatott képletekhez hasonló módszert használtam. Legyen $X$ az azonosításnál mért időket tartalmazó vektor, ami $N$ elemet tartalmaz, míg $R$ a regisztrálásnál létrehozott vektor, ami tartalmazza minden billentyűkombináció esetén a megfelelő idők mintabeli várható értékét ($\mu_i$) és annak mintabeli szórását ($\sigma_i$). Standardizáljuk az $X_i$ normál eloszlású valószínűségi vátozókat. Minden $X_i$ jellemzi az illető betűpárnál mért hasonlóságot és ha ezekből átlagot számolunk, akkor egy normál eloszlású mérőszámot kapunk. Ez a szám minél közelebb van a nullához, a hasonlóság annál nagyobb. Ekkor az azonosságot mérő függvény így néz ki:

$$D(X,R) = \frac{1}{N}\sum_{i=1}^{N}\frac{|X_i - \mu_{X_i}|}{\sigma_{X_i}} \tag{6}$$




1. ábra. Az **F** és **O** billentyűk környezete

Mivel összesen 66 vizsgált billentyő van és mindegyik kombinációt külön kell tárolni, összesen legalább 4356 értéket kellen regisztrálni. 3000 karakter erre semmiképpen nem elegendő, így módot kellett találni arra, hogy a hiányzó értékeket kitöltsem a meglévőek alapján. Tegyük fel, hogy, hogy az $X_{ij}$ elemnek szeretnénk megbecsülni a várható értékét ($\mu_{ij}$), illetve szórását ($\sigma_{ij}$). Legyen $K_i$ egy tetszőleges $i$ billentyű környezetében lévő billentyűk halmaza (például $K_f = \{R, T, D, G, C, V\}$ vagy $K_o = \{9, 0, I, P, K, L\}$, lásd 1. ábra). Továbbá legyen $U_{ij}^1 = \{x | x \in K_i, \mu_{xj} \neq 0\}$ és $U_{ij}^2 = \{x | x \in K_j, \mu_{ix} \neq 0\}$. Ekkor

$$\mu_{ij} = \frac{1}{|U_{ij}^1| + |U_{ij}^2|} \left( \sum_{k \in U_{ij}^1} \mu_{kj} + \sum_{k \in U_{ij}^2} \mu_{ik} \right) \tag{7}$$

illetve

$$\sigma_{ij} = \sqrt{\frac{1}{|U_{ij}^1| + |U_{ij}^2|} \left( \sum_{k \in U_{ij}^1} \sigma_{kj}^2 + \sum_{k \in U_{ij}^2} \sigma_{ik}^2 \right)} \tag{8}$$

Ezeket a kiegészítéseket először olyan esetekben végzem el, amikor $|U_{ij}^1| + |U_{ij}^2| > 7$, vagyis legalább 7 szomszédnak van regisztrált értéke. Minden lépésben csökkentem eggyel a megkövetelt szomszédok számát és a végén a ki nem egészített elemek helyére a ismert értékek átlagát írom. Ezzel az eljárással biztosítva van, hogy minden elem értéket kap.

# 4. Eredmények

A 1. táblázat tartalmaz néhány eredményt a szakirodalomból. Monrose kísérletében [3] 63 ember vett részt és a következő négyest használta az azonosításhoz: $M = \{M_{felhasznalonev},\ M_{jelszo},\ M_{keresztnev},\ M_{vezeteknev}\}$. Az adatok regisztrálása után, az

| **Módszer** | **FAR** % | **FRR** % |
|---|---|---|
| Euklideszi távolság [3] | 16.78% | |
| Euklideszi táv. hosszú szövegre [10] | 1.5% | |
| Nem súlyozott valószínűség [3] | 14.37% | |
| Súlyozott valószínűség [3] | 12.82% | |
| Bayes osztályozó [4] | 2.8% | 8.1% |
| SVM tetszőleges, rövid szövegre [11] | 0% | 3.54% – 15.78% |

1. táblázat. Eredmények a szakirodalomból

előbb említett módszerek közül hármat próbáltak ki, és ezekkel egyre jobb eredményeket értek el. A Bayes osztályozó használatakor 30 ember volt felkérve a gépelésre (15 ember rendelkezett regisztrált jelszóval és 15-nek csak behatoló feladata volt).

A hosszú szövegekre történő felismeréssel a Tappert [10] által végzett kísérlet próbálozott, itt csak 58 tulajdonságot vizsgáltak, amiről a 3.5 fejezetben volt szó. Az eredményekből látszik, hogy a hosszú szövegek hatékonyabbak, mint a rövid jelszavak.

Akkor tekinthetnénk igazán megbízhatónak ezt a típusú azonosítást, hogy ha az **FAR**-t 0-ra lehetne csökkenteni, vagyis idegent *soha* nem engedne be a rendszer. Ezt Yu [11] tűzte ki célul, és addig csökkentette az elfogadásnak a határértékét, amíg ezt el nem érte, de természetesen így többször utasít el jogos felhasználót.

Nem biztos, hogy helyes lenne a fenti számokat összehasonlítani, hiszen különböző szerzők különböző képpen értelmezhetik eredményeiket, sőt az SVM által előállított eredmény csalóka, hiszen ebben az esetben minden próbálkozó regisztrálta a tesztelt jelszót.

Attól függően, hogy a saját méréseimnél mennyire voltam szigorú a gépelések egyezésének az elfogadásánál, különböző eredmények születtek. Rövid jelszavak esetén ezeket a 2. táblázat tartalmazza. Látszik, hogy a billentyűk használata közötti idő jóval több információt hordoz írójáról, mint a billentyűk lenttartásának ideje, így jobban használható az azonosításhoz.

A hosszúszöveges mérés eredményeit a 3. táblázat tartalmazza. Ezek is függe-

| *Billentyűközök esetén* | |
|---|---|
| **FAR** | **FRR** |
| 25% | 10% |
| 17.7% | 16.4% |
| 5.57% | 27.1% |
| 0% | 48.4% |

| *Billentyűk lenttartása esetén* | |
|---|---|
| **FAR** | **FRR** |
| 66.66% | 6.88% |
| 24.56% | 22.5% |
| 5.26% | 36.88% |
| 0% | 55.63% |

2. táblázat. Saját eredmények rövid szövegek esetén

| *Billentyűk lenttartása esetén* | |
|---|---|
| **FAR** | **FRR** |
| 42.85% | 0% |
| 25% | 21.06% |
| 17.85% | 36.85% |
| 3.57% | 47.9% |

| *Billentyűközök esetén* | |
|---|---|
| **FAR** | **FRR** |
| 60.71% | 0% |
| 39.28% | 5.56% |
| 28.35% | 16.66% |
| 18.57% | 44.45% |

3. táblázat. Saját eredmények hosszú szövegek esetén

nek az egyezés szigorának a mághatározásától. Tetszőleges szövegre az eredmények kevésbé meggyőzőek, ennek elsődleges oka az, hogy a tesztszöveg mérete nem elég nagy, de ezt a méretet már nem nagyon lehet növelni, 3000 karakter így is eléggé sok. Érdekes megjegyezni, hogy a tesztelés során volt olyan *behatoló*, akinek a gépelési stílusa 98%-ban megegyezett a *tulajdonos* stílusával. Ez nemcsak a használt módszer esetleges gyengeségének köszönhető, hanem egyértelműen a két gépelési stílus hasonlóságának tulajdonítható be.

# 5. Adatok tárolása

Miután a regisztrálás folyamata közben létrejött egy *ujjlenyomatot* tartalmazó állomány, gondok lehetnek ennek tárolásával. Nem lehet egyirányú kódolást használni, mint általában a jelszavak esetén, hiszen a felhasználó nem tudja pontosan ezeket az időket reprodukálni minden belépéskor. Az adatok kódolatlan tárolása biztonsági

lyuk lenne, hiszen ebből, ha nem is egyértelműen, de visszanyerhető a felhasználó jelszava. Ezzel a problémával Monrose, Reiter és Werzel [1] foglalkozott. Az általuk kidolgozott megoldásban a jelszót, mint kulcsot használják az ujjlenyomatot tartalmazó állomány kódolásához. Így hozzájutva az újjlenyomathoz semmilyen információhoz nem jutunk sem a jelszóval, sem a gépelési időkkel kalcsolatban. Amennyiben ismerjük a jelszót azzal dekódolni tudjuk az ujjlenyomatot, de ha jobban belegondolunk, akkor ez nem is olyan nagy probléma. Az aláírást is elfogadjuk, mint hitelesítést, még akkor is, ha az írás eredményét bárki láthatja, ennek utánzása a nehéz feladat. Hasonlóan történik ebben az esetben is, hiába ismerjük a megfelelő időket, ezeknek az utánzása nem könnyű.

# 6. További tervek

A téma még messze nincs kimeríte, számos új megközelítést lehet kitalálni, kipróbálni. Az eddigi eredmények nem annyira meggyőzőek, hogy a valós életben is használni lehetne a megoldásokat. Eleddig még az is csak feltételezés, hogy a mért idők normál eloszlásúak, ez pedig a felhasznált képlet alapját képezi. Ennek bizonyítása körülményes, feloldása elérhetőbb célnak tűnik.

A végső cél az lenne, hogy kellően hosszú regisztráció után munka közben lehessen egyértelműen azonosítani a felhasználót. Ehhez nagyon sok adatra van szükség, aminek a beszerzése nem egyszrű, hiszen kevés ember engedi meg, hogy a háttérben működjön egy program amelyik minden billentyűleütést regisztrálja (a jelszavakat is). Saját gépemen működik egy, ami ennek a dolgozatnak a megírása közben is gyűjtötte az adatokat.

A téma beleillik a mesterséges intelligencia és adatbányászat által kutatott területekbe, a feladat megfogalmazható úgy, mint egy osztályozási feladat, annak eldöntésére, hogy aki gépel az a jogos felhasználó-e. Egy másik hasznos módszer lehet a komponens analízis, aminek segítségével le tudnánk csökkenteni az adatok dimenzióját, hiszen eléggé sok adat esetén több, mint 100 dimenzió vizsgálata erőforrásigényes. Ezeknek a módszereknek a kipróbálása izgalmas kihívás lehet.

# Hivatkozások


[1] F. Monrose, M. K. Reiter, and S. G. Wetzel: *Password hardening based on keystroke dynamics*, International Journal on Information Security 1(2):69-83, 2002.

[2] Fabian Monrosez, Aviel D. Rubin: *Keystroke Dynamics as a Biometric for Authentication* Future Generation Computer Systems, 2000.

[3] Fabian Monrose, Aviel D. Rubin: *Authentication via Keystroke Dynamics*, 4th ACM Conference on Computer and Communcations Security, 1997.

[4] Jarmo Ilonen: *Keystroke dynamics*, 2003, Elérhető: http://www.it.lut.fi/kurssit/03-04/010970000/seminars/Ilonen.pdf, Hozzáférve: 2007.

[5] F. Bergadano, D. Gunetti, C. Picardi: *Identity Verification through Dynamic Keystroke Analysis*, Intelligent Data Analysis (IDA), 7(5):469-496, 2003.

[6] Obaidat, M.S., Sadoun, B.: *Keystroke Dynamics BasedAuthentication* Elérhető: http://web.cse.msu.edu/ cse891/Sect601/textbook/10.pdf, Hozzáférve: 2007.

[7] Thomas, R.C., Karahasanovic, A., Kennedy, G.E.: *An Investigation into Keystroke Latency Metrics as an Indicator of Programming Performance*, In Proc. Seventh Australasian Computing Education Conference (ACE05), Newcastle, Australia. CRPIT, 42. Young, A. and Tolhurst, D., Eds., ACS. 127-134, 2005.

[8] Modi S. K., Elliott, S. J.: *Keystroke Dynamics Verification Using a Spontaneously Generated Password*, Proceedings of the 40th Annual International Carnahan Conference on Security Technology (ICCST) (pp. 116-121), 2006.

[9] Yingpeng Sang, Hong Shen, Pingzhi Fan: *Novel Impostors Detection in Keystroke Dynamics by Support Vector Machine*, Proc. of the 5th International Conference on Parallel and Distributed Computing: Applications and Technologies (PDCAT 2004),LNCS 3320, pp 666-669, Singapore, 2004.

[10] C.C. Tappert, M. Villani, M. Curtin, G. Ngo, J. Simone, H. St. Fort, and S. Cha: *Keystroke Biometric Recognition Studies on Long-Text Input over the Internet*, Proc. 23rd International Biometric Conf., Montréal, Canada, 2006.

[11] Enzhe Yu, Sungzoon Cho: *Keystroke Dynamics Identity Verification - its problems and practical solutions*, Computer and Security, Vol.23, No.5, pp. 428-440, 2004.

[12] C.J.C. Burges: *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, Vol. 2, Number 2, p. 121-167, Kluwer Academic Publishers, 1998.

[13] Horváth Gábor: *Bankbiztonság*, ELTE előadás, Elérhető: http://www.biztostu.hu/big_files/videok_es_hozzavalok/videok/Bank_HG.avi, Hozzáférve: 2007.

[14] Orvos Péter: *Biometria*, ELTE előadás, Elérhető: http://www.biztostu.hu/big_files/videok_es_hozzavalok/videok/Biometria_OP.avi, Hozzáférve: 2007.

[15] http://www.policensw.com/info/fingerprints/finger01.html

[16] R. Stockton Gaines, William Lisowski, S. James Press, Norman Shapiro: *Authentication by Keystroke Timing: Some Preliminary Results*, 1980.

[17] Rick Joyce, Gopal Gupta: *Identity authorization based on keystroke latencies.* Communications of the ACM, 33(2):168 176, 1990.