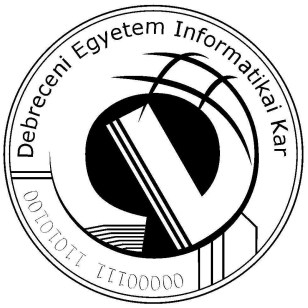
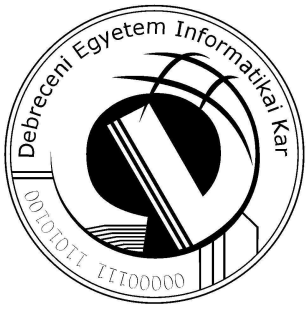



# Kriptográfiai hash függvények és álvéletlenszám generátorok

**Egyetemi doktori (PhD) értekezés**

**Folláth János**
Témavezető: Dr. Pethő Attila

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2011.




# Kriptográfiai hash függvények és álvéletlenszám generátorok

**Egyetemi doktori (PhD) értekezés**

**Folláth János**
Témavezető: Dr. Pethő Attila

Debreceni Egyetem
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2011.

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programja keretében készítettem a Debreceni Egyetem természettudományi doktori (PhD) fokozatának elnyerése céljából.
Debrecen, 2012. január 18.

.....
Folláth János
jelölt

Tanúsítom, hogy Folláth János doktorjelölt 2005 - 2012 között a fent megnevezett Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javasolom.
Debrecen, 2012. január 18.

.....
Dr. Pethő Attila
témavezető

# Kriptográfiai hash függvények és álvéletlenszám generátorok

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
az informatika tudományágban

Írta: Folláth János okleveles Programtervező Matematikus

Készült a Debreceni Egyetem Informatikai Tudományok doktori iskolája
Elméleti számítástudomány, adatvédelem és kriptográfia programja
keretében

Témavezető: Dr. Pethő Attila

A doktori szigorlati bizottság:

| | |
|---|---|
| elnök: | Dr. Végh János |
| tagok: | Dr. Györfi László |
| | Dr. Csirmaz László |

A doktori szigorlat időpontja: 2011. február 1.

Az értekezés bírálói:

| | | |
|---|---|---|
| | Dr. | ..... ..... |
| | Dr. | ..... ..... |

A bírálóbizottság:

| | | |
|---|---|---|
| elnök: | Dr. | ..... ..... |
| tagok: | Dr. | ..... ..... |
| | Dr. | ..... ..... |
| | Dr. | ..... ..... |
| | Dr. | ..... ..... |

Az értekezés védésének időpontja: 2012. .....

# Tartalomjegyzék

# 1. fejezet

# Bevezetés

## 1.1. Kriptológia

Napjainkban az informatika, a számítógépek át meg átszövik hétköznapi életünket. Ma a telefonjaink akkora számítási kapacitással rendelkeznek, mint harminc éve a teremnyi méretű számítógépek. Az internet, ami akkoriban a hadsereg játékszere volt, ma ott van minden háztartásban és a vezeték nélküli hálózatok segítségével mindenhová elkísér minket. A legkülönfélébb eszközök, mobiltelefonok, könyvolvasók és tablet PC-k biztosítják hozzáférésünket a hálózathoz, bárhol is legyünk. Ma egy átlagos gépkocsi nagyobb számítási kapacitással és diagnosztikai képességgel van ellátva, mint a hatvanas évek űrhajói. Szórakozásunk, munkánk és magánéletünk körül leomlanak a földrajzi határok, és ez minden előnye mellett veszélyeket is hordoz magában.

Csalók törekszenek arra, hogy adatainkat megszerezzék és azzal visszaélve anyagi haszonra tegyenek szert. Jelentsen ez akár olyan apró kényelmetlenséget, mint a kéretlen levelek, vagy a sokkal komolyabb kárt okozó virtuális zsebtolvajlást, azaz bankszámlánk megcsapolását. Rosszakaróink használhatják fel ellenünk azokat az adatokat, amelyeket csak barátainknak vagy kollégáinknak szántunk.

A pusztán elektronikus formában mozgó, nagy értékű információ nem

korlátozódik személyes szintre. Világvállalatok bonyolítanak nagy értékű tranzakciókat elektronikusan, érzékeny és értékes kutatás-fejlesztési rendszerek kapcsolódnak a hálózatra. Kormányzati és egészségügyi nyilvántartások, céges ügyféladatbázisok kínálnak vonzó célpontokat a tisztességtelen szándékú elemeknek.

Ezekre a fenyegetésekre a kriptográfia adja meg a választ. A kriptográfia teszi lehetővé, hogy a mobil informatika, a kiterjedt vezetékes és vezeték nélküli hálózatok, és az internet kényelmét nagy értékű tranzakciókra is felhasználhassuk, továbbá, hogy a 21. századi hétköznapi életben alapvető magánélethez fűződő jogaink biztonságban legyenek.

A számítógépekkel és a hálózatokkal együtt a kriptográfia is beszivárog életünk minden területére. A kriptográfiai protokollok teszik lehetővé az érzékeny adatok biztonságos továbbítását, a legkülönfélébb nagy értékű tevékenységek elvégzését. Segítségével elektronikusan írhatunk alá szerződéseket olyan biztonsággal, mintha csak a kedvenc tollunkkal tennénk, biztonságosan küldhetjük el e-mailjeinket és miatta nem kell attól tartanunk, hogy telefonbeszélgetéseinket lehallgatják. A kriptográfia teszi lehetővé a biztonságos vásárlást az interneten, és teremti meg az internetes bankolás és az elektronikus ügyintézés lehetőségét is (cégalapítás, adóbevallás, elektronikus számlák, védjegy és szabadalom bejegyzése stb.).

A kriptológia a titkosírás, a titkosítás tudománya. Két alkotóeleme a kriptográfia és a kriptoanalízis. A kriptográfia foglalkozik a titkosító algoritmusok tervezésével és konstruálásával, míg a kriptoanalízis a már létező módszerek elemzését, feltörését tűzi ki céljául.

## 1.2. Kriptográfiai primitívek

A kriptográfia hajdan kizárólag a titkos szervezetek és a hadseregek eszköze volt. Napjainkban az internet elterjedésével felhasználása teljesen általános lett. Az aszimmetrikus kriptográfia 1976-os felfedezésével [22] pedig a funkciója is messze túlmutat az eredeti alkalmazásán, nevezetesen, hogy $a$ pontból $b$ -be juttatunk biztonságosan egy üzenetet.

Habár az alkalmazások az elektronikus szavazáson át az anonim üzenet-

küldésig terjednek, a kriptográfiai szolgáltatásokkal kapcsolatban vannak közös elvárások [57]:

- *Titkosság.* A közölt információ csakis az illetékesek számára legyen hozzáférhető.
- *Integritás.* Csakis illetékesek legyenek képesek módosítani az információt. Ha illetéktelenek módosítják, az legyen észrevehető.
- *Hitelesítés.* Az adatok forrása vagy célja legyen hiteles. A szóban forgó hitelesítés nem feltétlenül jelenti a felek azonosságának felfedését, csupán annak bizonyítását, hogy a rendszer legitim felhasználói.
- *Letagadhatatlanság.* Senki sem tagadhat le a rendszerben vállalt kötelezettséget vagy elvégzett tevékenységet (például egy üzenet elküldését).

Ezeket a funkciókat a kriptográfiában egyes kriptográfiai primitívek segítségével érik el. A gyakorlatban a kriptográfiai primitíveket egymással összefüggésben, protokollokba szervezve használjuk, oly módon, hogy az együttes alkalmazásuk egy egészében biztonságos rendszert adjon.

A kriptográfiai primitíveket három nagy csoportra oszthatjuk kulcshasználatuk alapján.

### 1.2.1. Kulcs nélküli primitívek

A kulcs nélküli primitívek nem használnak kulcsokat, jellegüknél fogva az egyes kriptográfiai célokat biztosító sémák építőelemeiként használjuk őket. Ide tartoznak a

- Hash függvények
- Egyirányú permutációk
- Véletlen sorozatok

A hash függvényeket gyakran használjuk aláírási sémákban és a jelen dolgozat egyik fő témáját is egy új hash függvény konstrukció adja. Véletlen sorozatokat használunk például a szimmetrikus és a Vernam titkosítókban a kulcs szerepére. A Vernam titkosító működése során egy teljesen véletlen kulcsfolyamot használunk, amely éppen olyan hosszú, mint a titkosítandó üzenet. A kulcs bitjeit kizáró vagy művelettel adjuk hozzá az üzenet bitjeihez. A módszer hátránya, hogy alkalmazásához mindkét félnek birtokában kell lennie ugyanannak a véletlen kulcsnak, ami ráadásul olyan hosszú, mint maga az üzenet.

A véletlen sorozatok előállítása legtöbbször meglehetősen körülményes (drága és/vagy céleszközt igényel), az áthidaló megoldások pedig rendszerint rossz minőségű (nem teljesen véletlen) sorozatokra vezetnek. A véletlen és álvéletlen sorozatokról bővebben a 2. fejezetben lesz szó.

### 1.2.2. Szimmetrikus primitívek

A szimmetrikus (, más néven titkos kulcsú) kriptográfiai primitívek esetében a két kulcs vagy megegyezik, vagy pedig "könnyen" számolhatóak egymásból. Ilyenek a

- Szimmetrikus kulcsú kódolók
- Kulcsos hash függvények
- Szimmetrikus aláírási sémák
- Álvéletlen sorozatok
- Szimmetrikus azonosítási primitívek

A kódolók tovább bonthatóak blokk és folyamkódolókra. Folyamkódolóra példa a már említett Vernam titkosító, blokk kódolóra pedig az Advanced Encryption Standard [20] (AES) -ként szabványosított Rijndael titkosító. Ezek a hagyományos titkos üzenetküldést teszik lehetővé. Alkalmazásuk feltétele, hogy mindkét fél birtokában legyen a titkos kulcsnak.

Ide tartoznak még a kulcsos hash függvények, egyes digitális aláíró, illetve azonosító sémák. Ezen csoport képviselői az álvéletlen sorozatok is.

Álvéletlen sorozatokat használnak például sok folyamkódolóban. Ezeknél a Vernam kódolóhoz hasonlóan végzik a titkosítást, csak a valódi véletlen sorozatok helyett álvéletlen sorozatokat használnak. Jelen dolgozat egyik fő témáját is az álvéletlen sorozatok illetve ezek konstrukciója képezi (részletes tárgyalásuk a 3. fejezetben).

### 1.2.3. Aszimmetrikus primitívek

Az aszimmetrikus (, más néven publikus kulcsú) primitívek esetében két kulcsot használunk: publikus, illetve privát kulcsot. Az aszimmetrikus kifejezés arra utal, hogy a privát kulcs a publikusból nem számítható ki belátható időn belül a támadó rendelkezésére álló számítási kapacitással és algoritmusokkal. Aszimmetrikus primitívek a

- Aszimmetrikus kulcsú kódolók
- Aszimmetrikus aláírási sémák
- Aszimmetrikus azonosítási primitívek

Az aszimmetrikus kulcsú kódolók célja a szimmetrikus társaikhoz hasonlóan a titkosítás, ilyen például a népszerű RSA [62] titkosító. Az aláírási sémák a saját kezű aláírás elektronikus megfelelői és ezek teszik lehetővé az elektronikus írásbeliséget. Alkalmazásukban kulcsszerepet játszanak a már említett kriptográfiai hash függvények.

## 1.3. Egyirányú függvények

A kriptográfiában kulcsszerepet játszik az egyirányú függvények fogalma. A kriptográfiai primitívek fenti osztályozásából kitűnik, hogy egy részüknél a feladatuk ellátásához szükségünk van egy úgynevezett kulcsra. A megvalósítás jellegénél fogva ezek rendszerint adott hosszúságú bitsorozatokat jelentenek. Mindegyik kulcsos primitív törhető a kulcsok végigpróbálgatásával. Tehát az általuk nyújtott biztonság nem abszolút, csakis valamilyen számítási értelemben teljesítik a biztonsággal kapcsolatos követelményeket.

Az aszimmetrikus primitíveknél a két kulcs különbözőségét is ebben a számítási értelemben követeljük meg, azaz, hogy a publikus kulcsból a privát kulcsot csak nehezen lehessen kiszámítani. Itt megjelenik egy olyan igény, hogy az $f(x) : k_{publikus} \rightarrow k_{privát}$ függvény könnyen kiszámítható legyen, míg a függvény $g(x) : k_{privát} \rightarrow k_{publikus}$ inverzét már nehéz legyen kiszámítani. Ha nincs ilyen függvényünk, akkor nem is beszélhetünk aszimmetrikus kriptográfiáról.

A szóban forgó *könnyű*, illetve *nehéz* kiszámíthatóság fogalma a bonyolultságelmélet területére nyúlik át. Az egyes algoritmusokat aszerint osztályozzuk, hogy a bemeneti adatok hosszának növelésével milyen mértékben nő a számítási idő. Ennek alapján beszélhetünk logaritmikus, lineáris, polinomiális, illetve exponenciális idejű algoritmusokról. A könnyű kiszámíthatóság alatt a kriptográfiában legtöbbször a legfeljebb polinom idejű algoritmussal való számíthatóságot értjük, míg a nehéz kiszámíthatóságról beszélünk, ha a függvény csak legalább exponenciális idejű algoritmussal számolható ki.

Egyirányú függvényre egy szemléletes példa a diszkrét logaritmus vagy más néven index függvény [29].

**1.1. Példa.** *Ismert* $a, x \in \mathbb{Z}$ *és* $p$ *prím esetén*

$$a^x \equiv b \mod p$$

*hatványozás könnyen elvégezhető, de* $a, b \in \mathbb{Z}$ *és a* $p$ *prím ismerete esetén az* $x$ *kiszámítása már nehéz probléma.*

Az egyirányú függvények fogalma szoros kapcsolatban áll a kriptográfiai hash függvényekkel is, ezért nem csak a teljes aszimmetrikus kriptográfia szempontjából játszik fontos szerepet, de a jelen dolgozat témáját képező hash függvényeket is érinti. (Az egyirányú függvények szabatos definíciója a 4. fejezetben kapott helyet)

## 1.4. Áttekintés

A jelen dolgozat 3 nagyobb részből áll és fő témáját két kriptográfiai primitív jelenti. Az első rész a 2. és a 3. Fejezetből áll és egy álvéletlen generátor

konstrukcióval és a bináris sorozatokra vonatkozó álvéletlen mértékekkel foglalkozik. Itt első lépésben az álvéletlenség fogalmáról és klasszikus problémáiról lesz szó. Később a Sárközi és Mauduit által definiált álvéletlen mértékek ([55]) kerülnek ismertetésre. Sárközi és Mauduit maguk is adtak példát jó álvéletlenségi mértékkel rendelkező generátorra, és később, részben más szerzők közreműködésével, több jó konstrukció is született ([39], [49] [36] [37] [56] [34]). Ennek a résznek a végén egy saját generátor konstrukciója következik, amely mindamellett, hogy jó álvéletlen mértékekkel rendelkezik, szemben az összes korábbi konstrukcióval, kettő karakterisztikájú véges testeket vesz alapul.

A vizsgált álvéletlen generátor sok jó tulajdonsága bizonyítást nyer, úgy, mint jó álvéletlen mértékek, család bonyolultság és a szigorú lavinahatás tulajdonsága. Ezen tulajdonságok többsége a Sárközi és Mauduit által megkezdett kutatás nyomán lett definiálva és bizonyításuk saját eredmény. Az ebben a részben ismertetett eredmények a [23] és [25] cikkekben kerültek publikálásra.

Mindezek ellenére kiderül, hogy a vizsgált generátor nem alkalmas kriptográfiai felhasználásra, ugyanis a lineáris bonyolultsága túl alacsony. Ezzel együtt fény derül egy szoros kapcsolatra a lineárisan visszacsatolt léptetőregiszterekkel. Ez a kapcsolat gyors hardveres implementációt tesz lehetővé, és a jó statisztikai tulajdonságokkal együtt kiválóan alkalmassá teszi a generátort nem kriptográfiai alkalmazásokra.

A második rész a 4. fejezetből áll és egy kriptográfiai hash függvénnyel foglalkozik. A konstrukció kiküszöböli elődje hibáit ([6]), és az operandusok méretében is előrelépést jelent. Elméleti megfontolások alapozzák meg a függvény előképellenállóságát és egy jó statisztikai tulajdonsága is bizonyítást nyer. Ezen generátorra vonatkozó megállapítások Bérczes Attilával és Pethő Attilával közös eredményeink és a [7] cikkben kerültek publikálásra. A lavinahatásra vonatkozó tétel saját eredmény és eddig még nem lett publikálva.

A harmadik részt az 5. fejezet képezi. A szóban forgó UDHash kriptográfiai hash függvény gyakorlati megvalósításával és lehetséges alkalmazásával foglalkozik. Mint lehetséges alkalmazási terület, ismertetésre kerül a DESignIn azonosító rendszer is. A DESignIn tervezési megfontolásai kö-

zös eredményeink Huszti Andreával és Pethő Attilával és a [26] cikkben kerültek publikálásra. A gyakorlati vizsgálatok eddig publikálatlan saját eredmények és a B illetve a D Appendixben találhatóak.

# 2. fejezet

# Álvéletlen generátorok - Bevezetés

Ebben a fejezetben az álvéletlenség Sárközi és Mauduit által bevezetett mértékeiről lesz szó. Az első szakasz témája az álvéletlen generátorok elhelyezése a kriptográfiai primitívek között, illetve azok gyakorlati felhasználási szempontjai. A második szakasz az álvéletlenség fogalmát és klasszikus problémáit tárgyalja. A harmadik szakasz végén eljutunk a véletlenség klasszikus, Knuth által megadott definícióihoz ([45]), míg a negyedik szakaszban a Sárközi és Mauduit által definiált álvéletlen mértékek ([55]) kerülnek ismertetésre.

## 2.1. Álvéletlen generátorok a gyakorlatban

Az előző fejezetben vázolt osztályozásban a kriptográfiai primitívek között, a szimmetrikus kulcsú kategóriában szerepeltek az álvéletlen sorozatok. Az álvéletlen generátor (szabadon fogalmazva) az az algoritmus, amely álvéletlen sorozatokat állít elő.

Az álvéletlen generátorok szerepe a kriptográfiában sokrétű. Egyrészt, mint kulcsos kriptográfiai primitívet használják fel egyes sémákban és protokollokban. Ilyen alkalmazás például a kulcsfolyam generálása a Vernam

titkosító számára. Ebben az esetben a kulcs megléte és a sorozat újraelőállíthatósága fontos szerepet játszik. A kulcsot az álvéletlen generátorok esetében magnak (seed) nevezzük.

A másik tipikus alkalmazás, amikor egy valódi véletlen generátor helyett használjuk. Ilyen alkalmazás a legtöbb kulcsgenerálás, az inicializációs vektorok, a jelszófájlokban használt só (salt), a kódolás előtt az üzenet véletlenítésére használt bitek, és a kriptográfiai protokollokban előforduló egyszer használatos számok (nonce) is.

Ez a másik típusa az alkalmazásoknak, mintegy folytatása és továbbvitele a klasszikus álvéletlen generátoroknak. Ebben felhasználásban a mag nem kulcsként szerepel, hanem csupán a kiinduló véletlenszerűséget jelenti, amivel kiválasztunk egyet az álvéletlen sorozatok egy halmazából. A klasszikus álvéletlen generátorokat statisztikában és szimulációkban használják. Az ezektől megkövetelt tulajdonságok kimerülnek a reprodukálhatóságban és bizonyos statisztikai tulajdonságokban. A kriptográfiai protokollokban és sémákban gyakran előfordul, hogy véletlen értékekre van szükség, de a gyakorlati alkalmazásokban csak a legritkább esetben van lehetőség valódi véletlen generátorok alkalmazására. A gyakorlatban a lehetőségek csak kis mennyiségű, és gyakran egy támadó által manipulálható véletlen adat elérésére korlátozódnak. Az álvéletlen generátorok ilyen alkalmazása esetén a környezetből vett korlátozott mennyiségű véletlenszerűséget sokszorozzuk meg, hogy elegendő véletlen adat álljon rendelkezésre kriptográfiai alkalmazásaink számára.

A kezdeti véletlenséget az adott alkalmazástól függően a rendszer a legkülönfélébb helyekről szerezheti be. Kamerával rendelkező eszközökön zárt blendével készített képek, számítógépeken hálózati vagy felhasználói aktivitás, esetleg a lemezmeghajtókban keletkező turbulencia [21] is véletlenség forrásaként szolgálhatnak. A Linux operációs rendszer például véletlen adatokat gyűjt az eszközmeghajtókból, és a `/dev/random` eszközfájlon keresztül bocsájtja a programok rendelkezésére.

## 2.2. Álvéletlenségi tesztek

A klasszikus álvéletlen generátorokat elsősorban szimulációknál, illetve statisztikában alkalmazzák. Itt a statisztikai tulajdonságokon kívül semmilyen egyéb követelményt nem támasztunk álvéletlen sorozatainkkal szemben. Ezeket a generátorokat rendszerint különféle tesztekkel tették próbára és kezdetben a kriptográfiai generátorok esetében is a teszteket használtak erre a célra. Az álvéletlenség tesztelésének koncepciója fejlődött tovább a későbbiekben egy olyan szemléletté, ahol az egyszerű statisztikai tesztek helyébe aktív támadóalgoritmusok léptek.

A tesztek jellemzője, hogy semmit sem mondhatunk el azokról a generátorokról, amelyek átmennek a teszteken. (Természetesen azon kívül, hogy teljesítette a tesztet). Míg ha valamelyik elbukik, az mindenképpen gyengének minősül.

Míg kriptográfiában legtöbbször bináris álvéletlen sorozatokra van szükségünk, az álvéletlen generátorok eredeti területein, a szimuláció és az egyenletes eloszlás matematikai elméletében ez az igény nem élt. Ennek megfelelően három fő típusba sorolhatjuk az álvéletlen sorozatokat:

1. $E_N = e_1, \ldots, e_n,\ e_i \in [0, 1)$
2. $E_N = e_1, \ldots, e_n,\ e_i \in \mathbb{Z}_b$
3. $E_N = e_i, \ldots, e_n,\ e_i \in \{0, 1\}$.

Eredetileg a teszteket kifejezetten valamely csoport számára dolgozták ki, és nem mindegyik teszt alkalmazható mindegyik típusú sorozatra. Bizonyos esetekben a tesztek elméleti hátterének újragondolására van szükség a teszt kiterjesztéséhez.

A tesztek maguk alapvetően két típusba sorolhatóak aszerint, hogy az álvéletlen sorozatnak mely részeit vizsgálják. A teljes hosszon működő teszteket elméleti, míg a sorozatnak csupán egy szeletét vizsgáló teszteket gyakorlati teszteknek nevezzük. (Ez onnan ered, hogy a gyakorlatban használt biztonságos generátorok olyan hosszú sorozatokat generálnak, hogy azokat a teljes hosszukon vizsgálni praktikusan lehetetlen, így némely teszteknek való megfelelést elméleti úton kell bizonyítani.)

Az egyes tesztek feladata eredetileg annyi volt, hogy a sorozatok egyenletes eloszlását vizsgálják. Ebben a megközelítésben a legkézenfekvőbb teszt a $\chi^2$ próba, vagy az 1. esetben a Kolmogorov-Szmirnov próba alkalmazása. Jelen dolgozat témájának szempontjából jelentőséggel bír néhány korai álvéletlenségi teszt jellege. A következő fejezetben ugyanis, az ismertetésre kerülő álvéletlenségi mértékek ezen tesztek továbbgondolásának és általánosításának tekinthetőek. Knuth [45] több ilyen tesztet is felsorol:

- *Gyakoriságpróba.* Abból az elvárásból indul ki, hogy az álvéletlen sorozat egyenletes eloszlású legyen. Tulajdonképpen egy $\chi^2$ próba elvégzését jelenti, ahol a nullhipotézisünk az, hogy a sorozat egyenletes eloszlásból származik.

- *Sorozatpróba.* Nem a sorozat elemeit, hanem a sorozat egymást követő $n$ elemből álló blokkjainak sorozatát vizsgáljuk.

- *Hézagpróba.* Azt vizsgálja meg, hogy milyen hosszú elemsor van az egy adott intervallumba eső elemsorozatok között.

- *Pókerpróba.* A számsorozatban az egymást követő számötösöket vizsgáljuk és a bennük előforduló pókerminták (pár, két pár, drill, full, póker) alapján osztályozzuk őket.

- *Szelvénygyűjtőpróba.* Azt vizsgálja, hogy mekkora részsorozat szükséges ahhoz, hogy az összes lehetséges elem szerepeljen benne.

- *Permutációpróba.* A sorozatot $t$ hosszú blokkokra bontjuk, az elemeket pedig a nagyság szerinti sorrendben elfoglalt helyükkel címkézzük. Az ugyanazon permutációt képviselő blokkokat osztjuk egy csoportba és ez alapján végezzük el a próbát. Elsősorban az 1. típusú sorozatoknál szokás alkalmazni.

- *Futampróba.* A szerencsejátékok szimulációjánál használt véletlen generátorok által ihletett próba. Az eredeti sorozat monoton növekvő, illetve monoton csökkenő részsorozatainak hosszát vizsgálja.

- *"t-ből a legnagyobb" próba.* A sorozatot $t$ hosszú blokkokra bontjuk, mindegyikből a legnagyobbat vesszük, és az így kapott sorozatra alkalmazzuk a gyakoriságpróbát.

- *Ütközésvizsgálat.* Akkor alkalmazzuk, ha a véletlen elemek lényegesen több osztályba eshetnek, mint ahány megfigyelést végzünk, ekkor az elemek legtöbbje olyan osztályba esik, amelyben még nincs másik elem. Ha az elem mégis olyan osztályba esik, amelyikben már van másik elem, akkor ütközésről beszélünk. A próba során az ütközések számát figyeljük meg, és ha az nem túl nagy, akkor elfogadjuk a generátort.

- *Sorozatkorreláció-próba.* A sorozat elemeinek a megelőző elemektől való függését vizsgálja. $E_N = \{e_1, \ldots, e_N\}$ sorozat esetén a

  $$C = \frac{n(e_0e_1 + e_1e_2 + \ldots + e_{N-2}e_{N-1} + e_{N-1}e_0) - (e_0 + e_1 + \ldots + e_{N-1})^2}{n(e_0^2 + e_1^2 + \ldots + e_{N-1}^2) - (e_0 + e_1 + \ldots + e_{N-1})^2}$$

  statisztikát képezzük (gyakorlatilag az $e_j$ és $e_{j+1 \bmod n}$ sorozatok korrelációs együtthatóját kapjuk. Ez alapján a próba általánosítható, és képezhetjük a sorozatnak bármely ciklikus eltoltjával való korrelációs együtthatóját).

- *Részsorozatpróbák.* A sorozat minden $t$-edik eleméből képzett sorozatra alkalmaz valamilyen próbát.

## 2.3. A véletlenség fogalma

Természetesen az előző részben említetteken felül számtalan további teszt létezik, amellyel álvéletlen generátorokat tehetünk próbára. Egy generátor esetében nincs esélyünk mindegyiket elvégezni és az egyes teszteknek is megvannak a jellegükből fakadó hátrányaik. Nevezetesen, hogy az elméleti tesztek nem szűrik ki a lokális szabályszerűségeket, míg a gyakorlati teszteket csupán a generátor által előállított sorozatok töredékére, vagy a generátornak egy jelentősen leskálázott változatára lehet elvégezni.

Statisztikai és szimulációs eljárásoknál használt álvéletlen generátorok esetében a generátorok tesztelése viszonylag egyszerű: kiválasztják azokat a teszteket, amelyek az adott alkalmazásnál jelentőséggel bírnak, és egy olyan generátort alkalmaznak, amely a szóban forgó teszteken átmegy.

A kriptográfiában azonban ennél bonyolultabb a helyzet. Ahhoz, hogy az adott generátor biztonságos legyen, praktikusan minden teszttel (vagyis támadással) szemben ellenállónak kell lennie. Az eljárás kezdetben az volt, hogy egy-egy generátort, ami kellően sok tesztet teljesített, jónak fogadtak el. Ezután elkezdték használni, ami gyakorlatilag a tesztelés folytatását jelentette, immáron élesben.

Kézenfekvő az igény, hogy a generátorok konstrukciójánál ennél többet lehessen mondani az adott generátor jóságáról. Ezen igény kielégítésére két válasz is született. Ezek közül az első a kriptográfiai bizonyítható biztonság elméletének megközelítését alkalmazza a véletlenség fogalmára. A második pedig tulajdonképpen egy nagyon erős elméleti teszt, amely kvantitatív eredményével az egyes generátorok erősségének az összehasonlítását is lehetővé teszi. Ez utóbbi megközelítés a Sárközy és Mauduit által bevezetett ([55]) álvéletlenségi mértékek elmélete. (Ez az elmélet képezi a 2.4 szakasz témáját). A két technika a véges sorozatok véletlenségfogalmának két különböző megközelítéséből eredeztethető. A véletlenség fogalmának Knuth által bevezetett definíciói a szakasz hátralévő részében kerülnek ismertetésre.

A különböző típusú végtelen sorozatok véletlenségfogalmának vizsgálata a múlt század elejére nyúlik vissza. A Knuth [45] terminológiájával élve $\infty$-egyenletes sorozatok fogalmát Emile Borel [12] vezette be 1909-ben a 2-es típusú sorozatokra. A $\infty$-egyenletesség fogalma azóta ki lett terjesztve a többi típusra is. A végtelen sorozatok véletlenségfogalma köré azóta gazdag matematikai elmélet fejlődött. A $\infty$-egyenletesség fogalmának megadásához szükségünk lesz a $k$-egyenletesség fogalmára [45]:

**2.1. Definíció.** *Egy $E = \{e_0, e_1, e_2, \ldots\}$, $0 \leq e_i < b$ sorozatot $k$-egyenletesnek nevezünk, ha*

$$P(e_n = x_1, e_{n+1} = x_2, \ldots, e_{n+k-1} = x_k) = 1/b^k$$

*minden lehetséges* $x_1, \ldots, x_k$ *-ra ahol* $0 \leq x_i < b$.

A fenti definíció a 2-es típusú sorozatokra adja meg a $k$-egyenletesség fogalmát, és a további definícióink is a 2-es típusú sorozatokra fognak vonatkozni. Ugyan jelen dolgozat szempontjából a 3-as típusú sorozatok bírnak jelentőséggel, de vegyük észre, hogy a 3-as típusú sorozatok a 2-es típusúak speciális esetének is tekinthetők (nevezetesen a $b = 2$ eset).

**2.2. Definíció.** *Egy sorozatot* $\infty$*-egyenletesnek nevezünk, ha minden* $k$ *természetes számra* $k$*-egyenletes.*

A $\infty$-egyenletes sorozatok sok jó tulajdonsággal rendelkeznek és fontos szerepet játszanak a véletlenszerűség fogalmának leírásában. A $\infty$-egyenletes sorozatok megfelelnek az előző szakaszban leírt teszteknek. Ezek alapján egy kézenfekvő definíció a véletlenszerűségre a következő:

**2.3. Definíció** (Véletlenszerűség - I. jelölt)**.** *Egy sorozatot véletlenszerűnek nevezünk, ha az* $\infty$*-egyenletes.*

A $\infty$-egyenletesség azonban önmagában nem elegendő feltétele a véletlenszerűségnek, hiszen fogalmaink szerint egy végtelen $E = \{e_0, e_1, e_2 \ldots\}$ véletlen sorozattal kapcsolatban elvárnánk például, hogy a belőle képzett

$$E' = \{e_0, e_1, e_4, e_9, \ldots, e_{n^2}, \ldots\}$$

végtelen részsorozata is véletlen legyen. Ha azonban az $E$ sorozatot úgy módosítjuk, hogy $e_{n^2} = 0$, akkor a kapott sorozat továbbra is $\infty$-egyenletes marad, hiszen a $k$-egyenletességhez szükséges valószínűségek kiszámításához számlált $r_k(n)$ gyakoriság legfeljebb $b\sqrt{n}$-el változik. A $r_k(n)/n$ hányadosok határértéke így változatlan marad. Ennek alapján a következőképp is szigoríthatnánk a definíciót:

**2.4. Definíció** (Véletlenszerűség - II. jelölt)**.** *Egy* $E = \{e_0, e_1, e_2, \ldots\},\ 0 \leq e_i < b$ *sorozat véletlenszerű, ha minden végtelen részsorozata* $\infty$*-egyenletes.*

Ez a megfogalmazás azonban már túl erős lesz: egyetlen sorozat sem fog megfelelni neki. Minden egyenletes eloszlású $E = \{e_0, e_1, e_2, \ldots\},\ 0 \leq e_i <$

$b$ sorozatban végtelen sok 0 fog szerepelni. Ezeknek az indexeiből alkotott részsorozat nyilván nem $\infty$-egyenletes. Egy elfogadható definíció a kettő közötti kompromisszumból születik:

**2.5. Definíció** (Véletlenszerűség - III. jelölt)**.** *Egy $E = \{e_0, e_1, e_2, \ldots\}$, $0 \leq e_i < b$ sorozat véletlenszerű, ha minden kiszámítható részsorozatszabállyal kijelölt részsorozata 1-egyenletes.*

Ahol részsorozatszabálynak függvényeknek egy $f_n(x_1, \ldots, x_n)$ végtelen sorozatát fogjuk nevezni, ahol $f_n(x_1, \ldots, x_n)$ egy $n$ változós függvény, értéke 0 vagy 1, az $x_1, \ldots, x_n$ pedig valamely $S$ halmaz elemei. Ezen részsorozatszabály alapján az $e_n$ elem pontosan akkor eleme a részsorozatnek, ha $f_n(e_1, \ldots, e_n) = 1$. A részsorozatszabály kiszámítható, ha minden $f_n(x_1, \ldots, x_n)$ algoritmussal kiszámítható.

Ez a definíció már elfogadható a fenti szempontok szerint. Ugyan Knuth a későbbiekben további pontosításokat tett hozzá a fenti definícióhoz. Jelen dolgozat szempontjából azonban ez a definíció a legalkalmasabb, mert ez alapján egyszerűen meglátható a véletlenségfogalom kapcsolata az álvéletlenségi mértékekkel. (A következő szakaszban ismertetésre kerülő normalitás mérték például szoros kapcsolatban áll a $\infty$-egyenletesség fogalmával, a jóleloszlás és a korrelációs mértékekhez pedig felírható úgy részsorozatszabály, hogy annak a részsorozatnak az 1-egyenletességét mérje).

## 2.4. Álvéletlenségi mértékek

A véletlen sorozatok alkalmazásaiban természetesen véges sorozatokat alkalmazunk. A végtelen sorozatokra megadott definícióval analóg módon azonban a véges sorozatok véletlenségfogalmát is bevezethetjük. A fent vázolt $k$-egyenletességgel analóg definíciót alapul véve, a véges esetben alkalmazhatatlan $\infty$-egyenletesség helyett csupán a $k$-egyenletességet megkövetelve minden $k \leq \frac{\log N}{\log 2}$ esetén, a véges sorozatok egy, a 2.4 definícióval analóg véletlenfogalmára jutunk:

**2.6. Definíció.** *Egy* $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ *sorozat véletlenszerű, ha minden*

$$k \leq \frac{\log N}{\log 2}$$

$k \in \mathbb{N}$ *-re és minden* $X \in \{-1, 1\}^k$ *-ra*

$$\left| T(E_N, N + 1 - k, X) - \frac{N + 1 - k}{2^k} \right| \leq \frac{1}{\sqrt{N}}$$

*teljesül, ahol* $T(E, M, X) = |\{n : 0 \leq n < M, (e_{n+1}, \ldots, e_{n+k}) = X\}|$.

A véges sorozatok véletlenségfogalmára Knuth is ezt a definíciót adja a [45] könyvében.

Maudit és Sárközy [55] cikkükben ezt a megközelítést vitték tovább véges bináris sorozatok esetében. A szóban forgó cikkben céljuknak tűzték ki, egy olyan definíció megadását, ami a fentivel ellentétben, egy mérőszámot, egy mértéket is ad a vizsgált sorozat álvéletlenségére. További célkitűzés volt, hogy ezek a mértékek a gyakorlatban is alkalmazhatóak legyenek egyes álvéletlen sorozatra.

Egy ilyen tulajdonságokkal rendelkező mérték hasznos eszközként szolgál az egyes álvéletlen sorozatok összehasonlítására. A szerzők 3 kívánatos statisztikai tulajdonságot határoztak meg, aminek a teljesülését a mértéknek mérnie kell:

1. Normalitás

2. Jóleloszlás

3. Alacsony többszörös korreláció

Végtelen bináris sorozatok esetében ezeket a következőképp definiálhatjuk [55]: Minden $k \in \mathbb{N}, M \in \mathbb{N}, X = (x_1, \ldots, x_k) \in \{-1, 1\}^k, a \in \mathbb{Z}, b \in \mathbb{N}, D = (d_q, \ldots, d_k) \in \mathbb{N}^k, d_1 < \ldots < d_k$ esetén legyen

$$T(E, M, X) = |\{n : 0 \leq n < M, (e_{n+1}, e_{n+2}, \ldots, e_{n+k}) = X\}|,$$

$$U(E, M, a, b) = \sum_{j=1}^{M} e_{a+jb}$$

és

$$V(E, M, D) = \sum_{n=0}^{M-1} e_{n+d_1} e_{n+d_2} \dots e_{n+d_k}.$$

Ennek megfelelően a kívánt tulajdonságokat a következőképp adjuk meg:

**2.7. Definíció.** *Az $E_N = (e_1, e_2 \dots) \in \{-1, 1\}^\infty$ sorozat normális ha*

$$|T(E, M, X) - M/2^k| = o(M) \tag{2.1}$$

*minden $k$-ra és $X$-re $M \to \infty$ mellett.*

**2.1. Megjegyzés.** *Ennek a fogalomnak a véges változatát fogalmazza meg a 2.6 definíció. Összevetve a 2.2 definícióval, az is látható, hogy a normális sorozat egyben $\infty$-egyenletes is.*

**2.8. Definíció.** *Az $E_N = (e_1, e_2 \dots) \in \{-1, 1\}^\infty$ sorozat rendelkezik a jóleloszlás tulajdonságával, ha*

$$U(E, M, a, b) = o(M) \tag{2.2}$$

*minden $a$ -ra és $b$ -re $M \to \infty$ mellett.*

**2.9. Definíció.** *Az $E_N = (e_1, e_2 \dots) \in \{-1, 1\}^\infty$ sorozat többszörös korrelációja alacsony, ha*

$$V(E, M, D) = o(M) \tag{2.3}$$

*minden lehetséges $D$ -re $M \to \infty$ mellett.*

Végtelen sorozatokról lévén szó 2.1 és 2.3 ekvivalensek, illetve Niven és Zuckerman tétele alapján [59] a normalitásból következik a 2.2 tulajdonság is.

Véges sorozatok esetében azonban az összefüggés sokkal bonyolultabb, ezért ilyen esetben szükséges mindhármat külön kezelni és megkövetelni.

Mindhárom tulajdonságot átfogalmazhatjuk véges sorozatokra a normalitáshoz hasonló módon (2.6 definíció). Ez azonban még egy szokványos elméleti tesztre vezetne, amely csupán a megfelelt/nem felelt meg értékekkel látná el az egyes sorozatokat. A Mauduit és Sárközy által megadott definíciók azonban egy numerikus értékeket rendelnek az egyes sorozatokhoz is, ami által a "jó" sorozatok is összehasonlíthatók lesznek [55]:

**2.10. Definíció.** *Egy* $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ *sorozat* $k$*-ad rendű normalitásmértékén az*

$$N_k(E_N) = \max_{X \in \{-1,1\}^k} \max_{0 < M \leq N+1-k} |T(E_N, M, X) - M/2^k|$$

*értéket értjük. A sorozat normalitás mértéke ennek alapján:*

$$N(E_N) = \max_{k \leq (\log N)/\log 2} N_k(E_N).$$

**2.11. Definíció.** *Egy* $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ *sorozat jóleloszlás mértékén a*

$$W(E_N) = \max_{a,b,t} |U(E_N, t, a, b)| = \max_{a,b,t} |\sum_{j=1}^{t} e_{a+jb}|,$$

*ahol a maximumot minden olyan* $a, b, t$ *értékekre vesszük, ahol* $a \in \mathbb{Z},\ b, t \in \mathbb{N}$ *és* $1 \leq a + b \leq a + tb \leq N$*.*

**2.12. Definíció.** *Egy* $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ *sorozat* $k$*-ad rendű korrelációs mértékén a*

$$C_k(E_N) = \max_{M,D} |V(E_N, M, D)| = \max_{M,D} \left| \sum_{n=0}^{M-1} e_{n+d_1} e_{n+d_2} \ldots e_{n+d_k} \right|,$$

*ahol a maximumot minden olyan* $D = (d_1, \ldots, d_k)$ *és* $M$ *felett vesszük ahol* $M + d_k \leq N$*. A sorozat korrelációs mértéke ez alapján:*

$$C(E_N) = \max_{k \leq (\log N)/\log 2} C_k(E_N).$$

Míg végtelen esetben 2.1 és 2.3 ekvivalensek, addig a kapcsolat véges sorozatok esetén csupán egyirányú [55]:

**2.1. Lemma.** *Minden $N, E_N$ és $k < N$ esetén:*

$$N_k(E_N) \leq \max_{1 \leq t \leq k} |C_t(E_N)|.$$

A jóleloszlás és a korrelációs mérték között viszont nincs direkt összefüggés ezért aztán a szóban forgó véletlenszerűségfogalom mérőszámára a következő kombinált definíció adható [55]:

**2.13. Definíció.** *Egy $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ sorozat $k$-ad rendű álvéletlenségi mértékén a*

$$\begin{aligned} Q_k(E_N) &= \max_{a,b,t,D} \left| \sum_{j=0}^{t} e_{a+jb+d_1} e_{a+jb+d_2} \ldots e_{a+jb+d_k} \right| \qquad (2.4) \\ &= \max_{a,b,t,D} |Z(a,b,t,D)|, \end{aligned}$$

*ahol*

$$|Z(a,b,t,D)| = \left| \sum_{j=0}^{t} \sum_{j=0}^{t} e_{a+jb+d_1} e_{a+jb+d_2} \ldots e_{a+jb+d_k} \right| \qquad (2.5)$$

*kifejezést az olyan $a, b, t, D = (d_1, d_2, \ldots, d_k)$ -ra értjük amelyeknél minden $a+jb+d_l$ index $\{1, \ldots, N\}$ -ba tartozik és a (2.4) maximumot a $k$ dimenziós $D$ -k felett értjük.*

*Ennek alapján a sorozat álvéletlenségi mértéke:*

$$Q(E_N) = \max_{k \leq (\log N)/\log 2} Q_k(E_N).$$

# 3. fejezet

# Álvéletlenszám generátorok

Cikkükben ([55]) Sárközi és Mauduit maguk is adtak példát jó álvéletlenségi mértékkel rendelkező generátorra, és később, részben más szerzők közreműködésével, több jó konstrukció is született ([39], [49] [36] [37] [56] [34]). Az első szakaszban ezek közös tervezési elve, a dupla csavar módszer és az egyik, az új konstrukcióhoz közel álló generátor kerül ismertetésre. A második szakaszban az új generátor konstrukciója következik, amely mindamellett, hogy jó álvéletlen mértékekkel rendelkezik szemben az összes korábbi konstrukcióval, kettő karakterisztikájú véges testeket vesz alapul. Itt kerülnek bizonyításra az új konstrukció álvéletlen mértékei is. A kriptográfiában nem csupán álvéletlen sorozatokra, hanem nagy álvéletlen sorozat családokra van szükségünk. Fontos az is, hogy a sorozatok, ne csak külön, hanem együtt is jó tulajdonságokkal rendelkezzenek. Ilyen tulajdonság a család bonyolultság és a lavina hatás. Az új konstrukció ezen tulajdonságai a harmadik szakaszban nyernek bizonyítást. A negyedik szakaszban a generátor lineáris bonyolultságáról lesz szó. Kiderül, hogy nem alkalmas kriptográfiai felhasználásra, ugyanis a lineáris bonyolultsága túl alacsony. Ezzel együtt fény derül egy szoros kapcsolatra a lineárisan visszacsatolt léptetőregiszterekkel. Ez a kapcsolat gyors hardveres implementációt tesz lehetővé, és a jó statisztikai tulajdonságokkal együtt kiválóan alkalmassá teszi a nem kriptográfiai alkalmazásokra. A fejezet utolsó három szaka-

szának eredményei saját eredmények és a [23] és [25] cikkekben kerültek publikálásra.

## 3.1. Dupla-csavar módszer

Jó álvéletlenségi mértékekkel rendelkező generátorok megadásához a kettős csavar módszert fogjuk használni. Az egészek aritmetikai tulajdonságai és számjegyei közötti összefüggését vizsgáló kutatások [31] [28] [27] [54] [53] abba az irányba mutatnak, hogy ezek függetlenek. A módszer ezt a függetlenséget használja ki.

Első lépésben generálunk egy sorozatot. Könnyen lehet, hogy ez a számsorozat az aritmetikai tulajdonságai alapján nem felel meg a véletlenség fogalmainknak. Legtöbb konstrukció, amelyről bebizonyították, hogy jó álvéletlenségi mértékkel rendelkezik, például polinomiális kongruenciákat használ erre a célra.

A második lépésben a kettős csavar elvén működő álvéletlen generátorok lerombolják a szabályszerűséget hordozó aritmetikai struktúrát. Az így kapott sorozattól pedig valamilyen értelemben vett álvéletlen tulajdonságok teljesítését várjuk.

A módszer egyébként párhuzamba állítható a gyakorlatban használt, lineáris rekurzív sorozatok kombinálásával kapott álvéletlen generátorokéval. A lineáris rekurzív sorozatokon alapuló álvéletlen generátorok bizonyítottan jó statisztikai tulajdonságokkal rendelkeznek [43] [42] [45]. Ebből kiindulva az első lépést egy vagy több lineáris rekurzív sorozat adja, és második lépésben ezekre alkalmaznak egy nemlineáris szűrőt abban a reményben, hogy az eredményül kapott generátor megőrzi az lineáris rekurzív sorozatok jó statisztikai tulajdonságait, de nem rendelkezik azoknak a linearitásával és szabályszerűségeivel.

Egy dupla csavar elven működő álvéletlen generátort adtak meg mintának a szerzők [55] -ben jó álvéletlen mértékekkel rendelkező bináris sorozatokra. Legyen $p$ egy prímszám és $x$ egész az alábbiakban $\left(\frac{x}{p}\right)$ Legendre szimbólumot.

**3.1. Lemma.** *Legyen $p_0$ olyan, hogy $p > p_0$ prím. Legyen $k \in \mathbb{N}, k < p$ és*

*legyen*

$$E_{p-1} = \left(\left(\frac{1}{p}\right), \left(\frac{2}{p}\right), \ldots, \left(\frac{p-1}{p}\right)\right).$$

*Ekkor*

$$Q_k(E_{p-1}) \leq 9kp^{1/2}\log p.$$

A fenti konstrukcióval a legnagyobb probléma, hogy minden $p$ értékhez egyetlen sorozatot rendel. Az alkalmazásokban nagyszámú álvéletlen sorozat előállítására van szükség. Ezt a konstrukciót később permutációs polinomok segítségével terjesztették ki a probléma áthidalására [56]:

**3.1. Definíció.** *Az $f \in \mathbb{F}_q[x]$ polinom permutációs polinom, ha a hozzá tartozó $f : c \to f(c)$ polinomfüggvény az $\mathbb{F}_q$ egy permutációja.*

**3.2. Lemma.** *Legyen $p$ prím és legyen $g(x)$ egy $\mathbb{F}_p$ feletti $m$-ed fokú permutációs polinom. Legyen $g(x)$ olyan, hogy zérushelyeinek a multiplicitása páratlan. Definiáljuk az $E_p = \{e_1, e_2, \ldots, e_p\}$ sorozatot a következőképp:*

$$e_n = \begin{cases} \left(\frac{g(n)}{p}\right) & \text{ha } g(n) \not\equiv 0 \mod p \\ 1 & \text{ha } g(n) \equiv 0 \mod p \end{cases}.$$

*Ekkor minden $k \in \mathbb{N},\ k < p$ -re*

$$Q_k(E_p) < 11kmp^{1/2}\log p.$$

Később ez a konstrukció további általánosításra került [39], ezáltal polinomok egy szélesebb osztálya vált felhasználhatóvá. Az általánosításra vonatkozó eredmények kimondásához szükségünk lesz a következő ekvivalenciarelációra:

**3.2. Definíció.** *Azt mondjuk, hogy a $\varphi(x), \psi(x) \in \mathbb{F}_p[x]$ polinomok ekvivalensek:*

$$\varphi \sim \psi \tag{3.1}$$

*ha van olyan $c \in \mathbb{F}_p^*$ és $\gamma \in \mathbb{N}$, hogy $\varphi(x) = c\psi(\lambda^\gamma x)$.*

**3.3. Lemma.** *Legyen $p$ páratlan prím, $\lambda \in \mathbb{F}_p^*$ multiplikatív rendje $T$ és $f(x) \in \mathbb{F}_p[x]$ egy $k$-ad fokú polinom. Ekkor legyen $E_T = \{e_1, \dots, e_T\} \in \{-1, 1\}^T$, ahol*

$$e_n = \begin{cases} \left(\frac{f(\lambda^n)}{p}\right) & ha\ p \not| f(\lambda^n) \\ 1 & ha\ p | f(\lambda^n). \end{cases}$$

*Legyen $f(x)$ olyan, hogy nem írható fel $cx^\alpha(g(x))^2$ formában, ahol $c \in \mathbb{F}_p$, $\alpha \in \mathbb{N}$, $g(x) \in \mathbb{F}_p[x]$. Ekkor*

$$W(E_T) < 5kp^{1/2} \log p.$$

*Továbbá, ha $\beta \in \mathbb{N}$ a legnagyobb olyan egész, amelyre $x^\beta | f(x)$ teljesül, és a következő 4 feltétel valamelyike érvényes*

*a) $l = 2$, és $f(x)/x^\beta$ nem $g(x^\sigma)$ vagy $cx^\alpha(g(x))^2$ alakú, ahol $\sigma, \alpha \in \mathbb{N}, (\sigma, T) \geq 2, c \in \mathbb{F}_p$ és $g(x) \in \mathbb{F}_p[x]$.*

*b) $f(x)/x^\beta$ nem $cx^\alpha(g(x))^2$ alakú, ahol $\alpha \in \mathbb{N}$, $c \in \mathbb{F}_p$ és $g(x) \in \mathbb{F}_p[x]$, továbbá $T$ prím, és vagy $\min\{(4k)^l, (4l)^k\} \leq T$, vagy a 2 primitív gyök modulo $T$.*

*c) Tekintsük az $f(x)/x^\beta = \varphi_1^{\beta_1}(x) \dots \varphi_u^{\beta_u}(x)$ felbontást, ahol $\beta_i \in \mathbb{N}$ és $\varphi_i(x)$ irreducibilis $\mathbb{F}_p$ felett. Tegyük fel, hogy a 3.1 -beli $\sim$ reláció definiál egy olyan ekvivalencia osztályt, ami pontosan egy $\varphi_j$ $(1 \leq j \leq u)$ tényezőt tartalmaz $f(x)/x^\beta$ irreducibilis tényezői közül, továbbá ezen tényező multiplicitása $f(x)/x^\beta$ felbontásában $\beta_j = 1$.*

*d) $k - \beta$ és $l$ páratlanok.*

*Ekkor*

$$C_l(E_T) \leq 5klp^{1/2} \log p.$$

Ezen sorozatok további előnye, hogy elemeinek számítása lineáris rekurzióval is megoldható, ezáltal gyors implementációra nyílik lehetőség [24].

A jó álvéletlen mértékekkel rendelkező sorozatokra többféle konstrukció született (Például: [49] [36] [37] [56] [34]). Legtöbb ilyen sorozat esetében a

multiplikatív karakterek játszottak kulcsszerepet. Felmerül a kérdés, hogy additív karakterekhez kapcsolódó konstrukciók esetén milyen tulajdonságú sorozatok állnak elő. A következő eredmény egy olyan konstrukció álvéletlen mértékére ad korlátot, amelyben a prímtest feletti additív karakterekkel áll kapcsolatban [52]:

**3.4. Lemma.** *Legyen $p$ páratlan prím, $f(x) \in \mathbb{F}_p[x]$ egy $d$-ed fokú polinom, és definiáljuk az $E_p = \{e_1, \dots, e_p\}$ sorozatot a következőképp:*

$$e_n = \begin{cases} +1 & ha\ 0 \le r_p(f(n)) < p/2, \\ -1 & ha\ p/2 \le r_p(f(n)) < p \end{cases},$$

*ahol $r_p(n)$ azt az egyedi $r \in \{0, \dots, p-1\}$ értéket jelenti, amelyre $n \equiv r \pmod p$. Ekkor*

$$W(E_p) \ll dp^{1/2}(\log p)^2.$$

*Továbbá minden $2 \le l \le d-1$ esetén*

$$C_l(E_p) \ll dp^{1/2}(\log p)^{l+1}.$$

## 3.2. Álvéletlenségi mértékek

A korábban említett álvéletlen bináris sorozatok konstrukciójához minden esetben páratlan karakterisztikájú véges testeket használtak. Természetesen adódik a gondolat, hogy vajon páros karakterisztikájú testeken alapuló sorozatok milyen álvéletlen tulajdonságokkal rendelkeznek.

A következő, saját konstrukció is a dupla csavar módszer követi: felsorolja a test elemeit, majd pedig lerombolja az aritmetikai struktúrát. A páros karakterisztika mind az elemek felsorolása, mind a felhasznált karakterek területén korlátozza a lehetőségeket: az elemeket csak multiplikatívan lehet felsorolni, a multiplikatív karakterek használata pedig nem bináris sorozatot ad eredményül. Az előző szakaszban ismertetett korábbi konstrukciókhoz hasonlóan itt is polinomok használatával fog csak nagyobb mennyiségű álvéletlen sorozat előállni.

A tétel feltételeinek megfelelő struktúrák kiválasztása és a paraméterválasztás a 3.2.4 szakaszban kerül tárgyalásra.

**3.1. Tétel.** *Legyen $\mathbb{F}_q$ egy olyan kettő karakterisztikájú véges test, amelynek a multiplikatív csoportja prímrendű. Legyen $\chi$ egy additív nem fő karakter, és $\alpha$ az $\mathbb{F}_q$ primitív eleme, továbbá legyen $f(x) \in \mathbb{F}_q[x]$ fokszáma $c$ páratlan. Legyen $I$ az $f(x)$ nem nulla együtthatójú tagjai kitevőinek halmaza. Jelölje $\alpha^i$ minimálpolinomját $\mathbb{F}_2$ felett $m_i(x)$. Legyen*

$$E_{q-1} = \{\chi(f(\alpha^1)), \chi(f(\alpha^2)), \ldots, \chi(f(\alpha^{q-1}))\} \in \{-1, +1\}^{q-1}.$$

*Legyen $D' \subseteq \{1, \ldots, q-1\}$ olyan, hogy $\prod_{i \in I} m_i(x)$ nem osztja a $d(x) = \sum_{d_i \in D'} x^{d_i}$ polinomot, ekkor*

$$\max_{a,b,t,D'} |Z(a,b,t,D')| \leq 9dq^{1/2} \log q. \tag{3.2}$$

*Legyen $D \subseteq \{1, \ldots, q-1\}$ olyan, hogy $\prod_{i \in I} m_i(x)$ osztja a $d(x) = \sum_{d_i \in D} x^{d_i}$ polinomot, ekkor*

$$\max_{a,b,t,D} |Z(a,b,t,D)| = \max_{D}(q - 1 - \max_{d_i \in D} d_i). \tag{3.3}$$

**3.1. Megjegyzés.** *A $D = \{1, \ldots, k\}$ esetben, ahol $|I| > k$ az első rész oszthatósági feltétele és (3.2) teljesülnek.*

**3.2. Tétel.** *Legyen $\mathbb{F}_q$ kettő karakterisztikájú véges test ($q = 2^k$) és legyen a multiplikatív rendje prím. Legyen $\chi$ additív nem fő karakter, $\alpha$ pedig $\mathbb{F}_q$ primitív eleme. Legyen a $f(x) \in \mathbb{F}_q[x]$ fokszáma $d \geq \log q$ páratlan és az együtthatói pedig csak akkor legyenek nullák, ha az adott tag kitevője páros. Ha*

$$E_{q-1} = \{\chi(f(\alpha^1)), \chi(f(\alpha^2)), \ldots, \chi(f(\alpha^{q-1}))\} \in \{-1, +1\}^{q-1}, \tag{3.4}$$

*akkor:*

$$Q(E_N) \leq 9dq^{1/2} \log q. \tag{3.5}$$

**3.1. Folyomány.** *Ezek alapján*

$$W(E_N) \leq 9dq^{1/2} \log q$$

*nyilvánvalóan teljesül, továbbá minden* $l \leq d+1$ *-re*

$$C_l(E_N) \leq 9dq^{1/2} \log q.$$

### 3.2.1. Karakterösszegek

A fenti eredmények bizonyításához szükségünk lesz bizonyos karakterösszegekre vonatkozó felső korlátok ismeretére.

**3.5. Lemma** (Weil tétel)**.** *Legyen* $f \in \mathbb{F}_q[x]$ *fokszáma* $n \geq 1$ *és* $\gcd(n,q) = 1$*, továbbá legyen* $\chi$ *az* $\mathbb{F}_q$ *egy nem triviális karaktere. Ekkor*

$$\Big| \sum_{c \in \mathbb{F}_q} \chi(f(c)) \Big| \leq (n-1)q^{1/2}.$$

*Bizonyítás.* Ez az 5.38. Tétel a 223. oldalon a [48] könyvben. □

**3.6. Lemma.** *Ha* $m \in \mathbb{N}$*, a* $g(x) : \mathbb{Z} \to \mathbb{C}$ *függvény periodikus* $m$ *periódussal, továbbá* $X$ *és* $Y$ *valós számok, úgy hogy* $Y > 0$*, akkor*

$$\Big| \sum_{X < n \leq X+Y} g(n) \Big| \leq \frac{Y+1}{m} \Big| \sum_{n=1}^{m} g(n) \Big| + \sum_{1 \leq |h| \leq m/2} |h|^{-1} \Big| \sum_{n=1}^{m} g(n) e\Big(\frac{hn}{m}\Big) \Big|.$$

*Bizonyítás.* Ez az eredmény implicit a [73] könyvben, és explicit formájában [71] könyvfejezetben és [30] cikkben került közlésre és kapcsolatban áll az Erdős-Turán egyenlőtlenséggel. □

**3.7. Lemma.** *Legyen* $\chi$ *egy additív nem főkarakter és* $\alpha$ *a* $\mathbb{F}_q$ *véges test egy primitív eleme. Legyen* $h \in \mathbb{Z}$*,* $h \not\equiv 0 \bmod (q-1)$ *és az* $f(x) \in \mathbb{F}_q[x]$ *függvény* $d$ *fokszáma páratlan. Ekkor*

$$\Big| \sum_{n=1}^{q-1} \chi(f(\alpha^n)) e\Big(\frac{hn}{q-1}\Big) \Big| \leq dq^{1/2}.$$

*Bizonyítás.* Ez a [65] könyvben közölt 2G Tétel közvetlen következménye. □

Ennek segítségével megadhatjuk a felső korlátot a bizonyításban használt karakterösszegekre:

**3.3. Tétel.** *Tegyük fel, hogy $\chi$ egy additív nem főkarakter és $\alpha$ az $\mathbb{F}_q$ véges test egy primitív eleme, továbbá legyen az $f(x) \in \mathbb{F}_q[x]$ polinom $d$ fokszáma páratlan. Legyenek $X$ és $Y$ valós számok úgy, hogy $0 \le X < X+Y \le q-1$. Ekkor*

$$\Big| \sum_{X<n\le X+Y} \chi(f(\alpha^n)) \Big| < 9dq^{1/2}\log q.$$

*Bizonyítás.* A 3.6 Lemma alkalmazásával, az $m = q-1$ és $g(x) = \chi(f(\alpha^x))$ behelyettesítéssel:

$$\begin{aligned} \Big| \sum_{X<n\le X+Y} \chi(f(\alpha^n)) \Big| &\le \frac{Y+1}{q-1} \Big| \sum_{n=1}^{q-1} \chi(f(\alpha^n)) \Big| \\ &+ \sum_{1\le|h|\le\frac{q-1}{2}} |h|^{-1} \Big| \sum_{n=1}^{q-1} \chi(f(\alpha^n)) e\Big(\frac{hn}{q-1}\Big) \Big|. \end{aligned}$$

A 3.7 Lemma és a Weil tétel (3.5 Lemma) alkalmazásával a következőt kapjuk:

$$\begin{aligned} \Big| \sum_{X<n\le X+Y} \chi(f(\alpha^n)) \Big| &< 2dq^{1/2} + 2 \sum_{1\le h\le\frac{q-1}{2}} |h|^{-1} dq^{1/2} \\ &< 2dq^{1/2}(1 + (1 + \log\Big(\frac{q-1}{2}\Big))) < 2dq^{1/2}(2+\log q) \\ &\le 2dq^{1/2}\left(\frac{\log q}{\log 2} + \log q\right) < 9dq^{1/2}\log q \end{aligned}$$

és éppen ezt akartuk bizonyítani. □

### 3.2.2. Kódelmélet

A 3.2 Tétel bizonyításához szükségünk van néhány definícióra és eredményre a hibajavító kódok elméletéből is [50]:

**3.3. Definíció.** $\mathbb{F}_q[x]/(x^n-1)$ *egy ideálját* $n$ *hosszúságú ciklikus kódnak nevezzük.*

**3.8. Lemma.** *Legyen* $C$ *egy* $n$ *hosszúságú ciklikus kód. Ekkor*

- $C$*-ben egyértelműen létezik egy* $g(x)$ *minimális fokszámú polinom.*
- $C = \langle g(x) \rangle$*, azaz* $g(x)$ *a* $C$ *generátorpolinomja*
- $g(x)$ *osztja* $x^n - 1$ *-et*
- *Minden* $c(x) \in C$ *egyértelműen felírható* $c(x) = f(x)g(x)$ *alakban, ahol* $f(x) \in \mathbb{F}_q[x]$ *fokszáma kisebb, mint* $n-r$*, ahol* $r$ *a* $g(x)$ *fokszámát jelöli. A* $C$ *kód az* $f(x)$ *üzenethez az* $f(x)g(x)$ *kódszót rendeli.*

*Bizonyítás.* Ez az 1. Tétel a 190. oldalon a [50] könyvben. □

**3.4. Definíció.** *Az* $\boldsymbol{x} = x_1 \dots x_n$ *vektor (Hamming) súlya a vektorban szereplő nem nulla* $x_i$*-k száma.*

**3.5. Definíció.** *Az* $\boldsymbol{x} = x_1 \dots x_n$ *és* $\boldsymbol{y} = y_1 \dots y_n$ *vektorok közötti (Hamming) távolság azon pozíciók számát jelenti, amelyekben különböznek.*

**3.2. Megjegyzés.** *A ciklikus kódok esetén a kódszavakat nem vektorokkal, hanem polinomokkal reprezentáljuk, így a fenti definíciókban szereplő vektorok ebben az esetben a polinomok együtthatói által alkotott vektorok lesznek.*

**3.6. Definíció.** *A kód minimális távolsága, avagy a kódtávolság a kód kódszavai közötti minimális távolságot jelenti.*

**3.9. Lemma** (BCH korlát)**.** *Legyen* $C$ *egy ciklikus kód* $g(x)$ *generátorpolinommal úgy, hogy valamely* $b \geq 0, \delta \geq 1$ *egészekre és* $\mathbb{F}_q$ *egy* $\alpha$ *primitív elemére*

$$g(\alpha^b) = g(\alpha^{b+1}) = \ldots = g(\alpha^{b+\delta-2}) = 0$$

*(BCH kód). Ekkor a* $C$ *kód távolsága legalább* $\delta$*.*

*Bizonyítás.* Ez a 8. Tétel a 201. oldalon a [50] könyvben. □

### 3.2.3. Az eredmények bizonyítása

*A 3.1 tétel bizonyítása.* Legyen $Z(a,b,t,D)$ a (2.5) egyenlet szerint definiálva. Ekkor $k < q-1$ esetén

$$|Z(a,b,t,D)| = \left| \sum_{n=0}^{t} \chi(f(\alpha^{a+nb+d_1}))\chi(f(\alpha^{a+nb+d_2}))\dots\chi(f(\alpha^{a+nb+d_k})) \right|$$

minden $a,b,t,D=(d_1,\dots,d_k)$ értékre, amelyre

$$a+nb+d_l \in \{1,\dots,q-1\} \text{ ahol } n=0,1,\dots,t \text{ and } l=1,\dots,k. \quad (3.6)$$

Tegyük fel, hogy $f(x)=\sum_{i=0}^{c} a_i x^i$. Vezessük be a következő jelöléseket: $a_{ij}=a_i\alpha^{i(a+d_j)}$ és $f_j(x)=\sum_{i=0}^{c} a_{ij}x^i$. Ekkor

$$|Z(a,b,t,D)| = \left| \sum_{n=0}^{t} \chi(f_1(\alpha^{nb}))\chi(f_2(\alpha^{nb}))\dots\chi(f_k(\alpha^{nb})) \right| = \left| \sum_{n=0}^{t} \chi(\tilde{f}(\alpha^{nb})) \right|,$$

ahol $\tilde{f}(x)=\sum_{i=0}^{c} a'_i x^i$, $a'_i=\sum_{j=0}^{k} a_{ij}=\sum_{j=0}^{k} a_i(\alpha^i)^{a+d_j}$. Legyen $\tilde{d}(x)=x^a d(x)=\sum_{j=0}^{k} x^{a+d_j}$, ekkor $a'_i=a_i\tilde{d}(\alpha^i)$. $\tilde{d}(x)$ kifejezést tekinthetjük $\mathbb{F}_2$ feletti polinomnak is. Ebből az következik, hogy $\tilde{d}(\alpha^i)$ pontosan akkor lesz nulla, ha az $\alpha^i$ elem $m_i(x)$ minimálpolinomja osztja a $\tilde{d}(x)$ polinomot. Mivel $m_i(x)\neq x$ és irreducibilis, pontosan akkor fogja a $\tilde{d}(x)$ polinomot osztani, ha a $d(x)=\sum_{j=0}^{k} x^{d_j}$ polinomot is osztja. Következésképp $\tilde{f}(x)$ akkor és csakis akkor lesz nullpolinom, ha $\prod_{i:a_i\neq 0} m_i(x)$ osztja $d(x)$-et. Ebben az esetben a következőt kapjuk:

$$|Z(a,b,t,D)| = \left| \sum_{n=0}^{t} \chi(\tilde{f}(\alpha^{nb})) \right| = t.$$

Mivel a (3.6) feltétel érvényes és $t \le q-1-\max_{1\le l\le k} d_l$ ez bizonyítja a (3.3) állítást.

Ha $\prod_{i:a_i\neq 0} m_i(x)$ nem osztja a $d(x)$ polinomot, akkor a $\beta = \alpha^b$ jelöléssel a következőre jutunk:

$$|Z(a,b,t,D)| = \left| \sum_{n=0}^{t} \chi(\tilde{f}(\beta^n)) \right|.$$

Mivel $\mathbb{F}_q^*$ prímrendű, $\beta$ pontosan akkor lesz a $\mathbb{F}_q$ primitív eleme, ha $b \neq q-1$. Ha $b = q-1$ akkor $b = 1$, $a = d_1 = 0$, és ezért $|Z(a,b,t,D)| = |\chi(f(\alpha^{a+nb+d_1}))| = 1$ és a tétel teljesül. Egyébként alkalmazhatjuk a 3.3 Lemmát, amivel azt kapjuk, hogy

$$|Z(a,b,t,D)| = \left| \sum_{n=0}^{t} \chi(\tilde{f}(\beta^n)) \right| \leq 9dq^{1/2}\log q.$$

Ez bizonyítja a (3.2) állítást és ezzel a 3.1 Tétel bizonyítása teljes. □

*A 3.2 tétel bizonyítása.* Legyen $g(x) = l.c.m.\{m_1(x), m_3(x), \ldots, m_c(x)\}$. Ekkor $g(x)$ egy $C$ BCH kód generátorpolinomja $\mathbb{F}_q$ felett. Ekkor a BCH korlát alapján (3.9 Lemma) a $C$ kód minimális távolsága legalább $c+1$. Legyen $M(x) = \prod_{i=0}^{\frac{c-1}{2}} m_{2i+1}(x)$. Mivel $g(x)|M(x)$, $M(x)$ akkor és csakis akkor osztja a $d(x) \in \mathbb{F}_2[x]$ polinomot, ha $d(x)$ egy kódszó. Ebből következik, hogy $M(x)$ csak olyan polinomokat oszthat, amiknek a súlya nagyobb, mint $c$. Ebből a 3.1 Tétel alkalmazásával kapjuk az állítást. □

**3.3. Megjegyzés.** *A bizonyításban követett gondolatmenet a $f(x) = \sum_{i=0}^{c} a_i x^i$ polinomra vonatkozó feltételek kis módosításával is alkalmazható. Ha van olyan $a$ egész, hogy minden $j = a, \ldots, a + \log q - 1$ értékhez létezik a $\alpha^j$ elemnek olyan $\alpha^{c_j}$ konjugáltja, hogy $a_{c_j} \neq 0$, akkor a (3.5) állítás érvényes marad.*

### 3.2.4. A konstrukcióval kapcsolatos megjegyzések

A test megválasztását a 3.1 Tétel erősen korlátozza: a tétel állításai csak akkor érvényesek, ha a $q-1$ Mersenne prím. Ez erősen korlátozza a gyakorlati alkalmazásokban használható jelöltek számát: jelenleg mindössze

47 Mersenne prímet ismerünk (az éppen aktuális ismert Mersenne prímek megtekinthetőek a [35] weboldalon). Éppen ezért az álvéletlen generátor megvalósításánál érdemes egy, az adott alkalmazáshoz leginkább alkalmas prímet (és ahhoz tartozó véges testet) választani és a sorozatcsaládot pedig az egyéb paraméterek segítségével definiálni. Ezt alapvetően három különböző módon tudjuk véghez vinni:

1. Változó $\chi$, változatlan $\alpha$ és $f(x)$
2. Változó $\alpha$, változatlan $\chi$ és $f(x)$
3. Változó $f(x)$, változatlan $\alpha$ és $\chi$

Az első két esetben a jelenlegi biztonsági követelmények mellett a test méretét érdemes a 12. ($k = 127$) vagy nagyobb ($k = 521$ és $k = 607$) Mersenne prímhez megválasztani annak érdekében, hogy a kimerítő kulcstámadásnak ellenálljon. A harmadik lehetőség jóval nagyobb rugalmasságot és több lehetőséget nyújt az álvéletlen sorozatok megtervezése során.

Az íly módon nyert álvéletlen generátor jó álvéletlen mértékekkel rendelkezik. Azonban mivel természeténél fogva az álvéletlen mérték több elméleti teszt számszerűsítését jelenti, ez önmagában nem jelenti azt, hogy a generátor kriptográfiai szempontból is biztonságos. A generátor egyéb tulajdonságairól, és a kriptográfiai biztonsági megfontolásokról a következő szakaszban lesz szó.

## 3.3. Család tulajdonságok

Goubin, Mauduit és Sárközy tanulmányozták álvéletlen bináris sorozatok egy új, nagy családját [34]. Ez a konstrukció szintén a [55] cikkben ismertetett sorozatnak egy kiterjesztése. Habár biztonsága nem bizonyítható redukciós módszerekkel, sok matematikai érv szól mellette: rendelkezik a szigorú lavinahatás tulajdonságával [72], család bonyolultsága magas [4], a legjobb ismert támadások számítási bonyolultsága magas, létezik gyors implementációja [44], és a korrelációs mértékére vonatkozó korlátok lehetővé teszik, hogy a lineáris bonyolultság profilját becsüljük [13][5]. Mindezeket figyelembe véve a generátor biztonsága matematikailag megalapozott

és gyorsabb implementációval rendelkezik, mint legtöbb bizonyíthatóan biztonságos generátor.

A [34] cikkben említett generátor prímtestek elemeit és multiplikatív karaktereit használja. Ezzel szemben az előző szakaszban ismertetett generátor kettő karakterisztikájú testek elemeivel és additív karakterekkel dolgozik.

Ugyancsak az előző szakasz alapján ennek a generátornak is jó az álvéletlen mértéke, de felmerül a kérdés, hogy az egyéb tulajdonságai, úgy mint lavina hatás, család bonyolultság, lineáris bonyolultság profilja mennyire kedvezőek. A továbbiakban az előző szakaszban ismertetett generátor ezen tulajdonságait fogjuk megvizsgálni. Ehhez szükségünk lesz az álvéletlen sorozatok családjának a fogalmára:

**3.7. Definíció.** *Legyen $N \in \mathbb{N}$, $S$ pedig egy adott halmaz, minden $s \in S$ értékhez legyen hozzárendelve egy egyedi bináris sorozat:*

$$E_N = E_N(s) = (e_1, \ldots, e_N) \in \{-1, 1\}^N.$$

*Jelölje $F = F(S)$ az álvéletlen bináris sorozatoknak a családját:*

$$F = F(S) = \{E_N(s) : s \in S\}. \tag{3.7}$$

### 3.3.1. Lavina hatás

A lavinahatás tulajdonsága jól ismert a blokk-kódolók elméletében. Ez a tulajdonság azt az elvárást fejezi ki, hogy a kódoló bemenetén megváltoztatott egyetlen bit minden egyes kimeneti bitet 1/2 valószínűséggel változtasson meg.

A [72] cikkben a szerző a lavina hatás fogalmát adaptálja álvéletlen bináris sorozatokra. Ebben a szakaszban be fogjuk bizonyítani, hogy a szóban forgó álvéletlen generátor rendelkezik a szigorú lavinahatás tulajdonságával.

A lavinahatás definiálásához meg kell adnunk az álvéletlen bináris sorozatok közötti távolság fogalmát:

**3.8. Definíció.** *Ha* $N \in \mathbb{N}$ , $E_N = (e_1, \ldots, e_N) \in \{-1, 1\}^N$ *és* $E'_N = (e'_1, \ldots, e'_N) \in \{-1, 1\}^N$, *akkor definiáljuk a* $d(E_N, E'_N)$ *távolságot a következőképp:*

$$d(E_N, E'_N) = |\{n : 1 \leq n \leq N, e_n \neq e'_n\}|.$$

Ez tulajdonképpen a két sorozat közötti Hamming-távolsággal egyezik meg. Az álvéletlen bináris sorozatokra vonatkozó lavinahatás tulajdonság definíciója ennek fényében:

**3.9. Definíció.** *Bináris sorozatok egy* $F(S)$ *családja rendelkezik a szigorú lavinahatás tulajdonsággal, ha*

$$m(F) = \min_{\substack{s,s' \in S \\ s \neq s'}} d(E_N(s), E_N(s')) \geq \left(\frac{1}{2} + o(1)\right) N.$$

A lavinahatás tulajdonság vizsgálatakor az előző szakasz generátorának a paraméterezéséhez a 3. módszert fogjuk alkalmazni, azaz a különböző sorozatokat ugyanazon karakter és generátor elem segítségével generáljuk, az egyes sorozatokat az előállításhoz használt polinom határozza meg. Ezzel kapcsolatban természetesen adódik a polinomoknak egy osztálya, amelynek minden eleme megfelel a 3.1 Tétel feltételeinek (ezáltal csupa jó álvéletlen mértékkel rendelkező sorozat tartozik az elemeihez) és számítási szempontból is kezelhetőek. A polinomoknak ezt az osztályát a továbbiakban fésűs polinomoknak fogjuk nevezni.

**3.10. Definíció.** *Ha egy* $f(x) \in \mathbb{F}_q[x]$ *polinom* $f(x) = \sum_{i=0}^{d} a_i x^{2i+1}$ *formájú, akkor fésűs polinomnak nevezzük.*

**3.4. Tétel.** *Legyen* $S$ *a legfeljebb* $d$ *fokú* $f(x) \in \mathbb{F}_q$ *fésűs polinomok halmaza. Definiáljuk az egyes* $E_{q-1} = E_{q-1}(f) = \{e_1, \ldots, e_{q-1}\}$ *sorozatokat (3.4) szerint, az* $F = F(S)$ *családot pedig (3.7) alapján. Ha* $d = o(q^{1/2})$ *teljesül, akkor az* $F$ *sorozatcsalád rendelkezik a szigorú lavinahatás tulajdonsággal.*

*Bizonyítás.* Legyenek $f_1(x), f_2(x) \in S$ különbözőek. Ekkor $f_1(x) + f_2(x)$ nyilván nem nullad-fokú, és ezért a Weil tétel alapján (3.5 Lemma) következik, hogy:

$$|q - 1 - 2d(E_{q-1}(f_1), E_{q-1}(f_2))| = \left|\sum_{i=1}^{q-1} \chi(f_1(\alpha^i))\chi(f_2(\alpha^i))\right| \leq (d-1)q^{1/2}.$$

Így, ha $q - 1 - 2d(E_{q-1}(f_1), E_{q-1}(f_2)) \geq 0$, akkor:

$$\left(\frac{1}{2} - \frac{(d-1)q^{1/2}}{q-1}\right)(q-1) \leq d(E_{q-1}(f_1), E_{q-1}(f_2)) \leq \frac{q-1}{2} \tag{3.8}$$

érvényes. A $q - 1 - 2d(E_{q-1}(f_1), E_{q-1}(f_2)) < 0$ esetben a

$$\frac{q-1}{2} < d(E_{q-1}(f_1), E_{q-1}(f_2)) \leq \left(\frac{1}{2} - \frac{(d-1)q^{1/2}}{q-1}\right)(q-1) \tag{3.9}$$

egyenlőtlenség triviálisan következik.

Ekkor (3.8) és (3.9) alapján

$$\left(\frac{1}{2} - \frac{(d-1)q^{1/2}}{q-1}\right)(q-1) \leq m(F) \leq \left(\frac{1}{2} + \frac{(d-1)q^{1/2}}{q-1}\right)(q-1)$$

következik. □

### 3.3.2. Család bonyolultság

A [4] cikkben a szerzők egy új mértéket vezettek be, amely már nem egyes álvéletlen bináris sorozatokra, hanem sorozatcsaládokra vonatkozik. Az f-bonyolultság fogalmát egy gyakorlati probléma ihlette: tekintsük azt az esetet, amikor az álvéletlen generátort egy folyamkódoló kulcsfolyamának a generálására használjuk. Ilyen esetekben a támadó a kódolatlan üzenetek szabályszerűségei alapján meg tudja tippelni az üzenet egyes bitjeit és ebből a kulcsfolyam vonatkozó bitjeit is. Ezen ismeretek matematikai modellje lesz a specifikáció fogalma:

**3.11. Definíció.** *Definiáljuk a $j$ hosszú specifikációt egy $(i_1, \ldots, i_j)$ index halmaz és egy hozzá tartozó $(\varepsilon_{i_1}, \ldots, \varepsilon_{i_j}) \in \{+1, -1\}^j$ érték halmaz együtteseként. Azt mondjuk, hogy egy $\{e_1, \ldots, e_N\}$ bináris sorozat kielégíti a specifikációt, ha*

$$e_{i_1} = \varepsilon_1, \ldots, e_{i_j} = \varepsilon_j.$$

Az, hogy egy adott hosszúságú specifikációnak hány sorozat felel meg a vizsgált sorozatcsaládban, összefüggésben van a vonatkozó generátor a fent vázolt szituációban mutatott erejével, biztonságával. Ezt a jellemzőt hivatott jelezni a család bonyolultság:

**3.12. Definíció.** *Az $E_N \in \{-1, +1\}^N$ bináris álvéletlen sorozatok egy $F$ családjának $\Gamma(F)$ f-bonyolultsága a legnagyobb $j$ egész, amelyre teljesül, hogy minden $j$ hosszú specifikációhoz van legalább egy $E_N \in F$, amely kielégíti.*

A generátor $f$-bonyolultságának vizsgálatához szükségünk lesz a következő eredményekre:

**3.10. Lemma.** *A $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ polinom pontosan akkor permutációs polinom $\mathbb{F}_q$ felett, ha*

$$\sum_{(c_1,\ldots,c_n)\in\mathbb{F}_q^n} \chi(f(c_1, \ldots, c_n)) = 0$$

*$\mathbb{F}_q$ minden nem triviális $\chi$ additív karakterére.*

*Bizonyítás.* Ez a 7.38. következmény a [48] könyvben. □

**3.11. Lemma.** *Tegyük fel, hogy a $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ polinom*

$$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_m) + h(x_{m+1}, \ldots, x_n),\ 1 \leq m < n$$

*formájú. Ha $g$ és $h$ közül legalább az egyik permutációs polinom $\mathbb{F}_q$ felett, akkor $f$ is permutációs polinom $\mathbb{F}_q$ felett.*

*Bizonyítás.* Ez az állítás része a 7.42. tételnek a [48] könyvben. □

**3.12. Lemma.** *Tegyük fel, hogy az $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ polinom*

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{n} \alpha_i x_1$$

*formájú. Ha legalább egy $\alpha_i$ nem nulla, akkor $f$ permutációs polinom $\mathbb{F}_q$ felett.*

*Bizonyítás.* Az állítás indukcióval következik a 3.11 Lemmából és abból a tényből, hogy $g(x) = \alpha x$ permutációs polinom. □

A következő tétel alapján a vizsgált bináris álvéletlen sorozatcsalád ebből a szempontból is jó tulajdonságokat mutat.

**3.5. Tétel.** *Legyen $S$ a legfeljebb $d$ fokú $f(x) \in \mathbb{F}_q$ fésűs polinomok halmaza. Definiáljuk az egyes $E_{q-1} = E_{q-1}(f) = \{e_1, \ldots, e_{q-1}\}$ sorozatokat (3.4) szerint, az $F = F(S)$ családot pedig (3.7) alapján. Ha $A$ egy $t \leq \lfloor \frac{d+1}{2} \rfloor$ tagból álló specifikáció és $G(A)$ a $F = F(S)$ család azon részhalmazát jelenti, amelyek az $A$ specifikációt kielégítik, akkor*

$$|G(A)| = \frac{|F|}{2^t}$$

*teljesül.*

*Bizonyítás.* Legyen $A$ egy $t$ tagú specifikáció, amely a $(\varepsilon_1, \ldots, \varepsilon_t) \in \{+1, -1\}^t$ értékhalmaz és az $(i_1, \ldots, i_t)$ index halmaz együttese, ahol $1 \leq i_1, < \ldots <$

$i_t \leq q-1$. Ekkor

$$\begin{aligned}
|G(A)| &= \sum_{E(f)\in F}\prod_{j=1}^{t}\frac{(e_{i_j}+\varepsilon_j)\varepsilon_j}{2} = \frac{\varepsilon_1\dots\varepsilon_t}{2^t}\sum_{E(f)\in F}\prod_{j=1}^{t}(e_{i_j}+\varepsilon_j)\\
&= \frac{\varepsilon_1\dots\varepsilon_t}{2^t}\sum_{E(f)\in F}\left(\sum_{r=0}^{t-1}\sum_{1\leq j_1<\dots<j_r\leq t}\varepsilon_{j_1}\dots\varepsilon_{j_r}\prod_{\substack{1\leq s\leq t\\ s\notin\{j_1,\dots,j_r\}}}e_{i_s}+\varepsilon_1\dots\varepsilon_t\right)\\
&= \frac{|F|}{2^t}+\frac{\varepsilon_1\dots\varepsilon_t}{2^t}\sum_{E(f)\in F}\sum_{r=0}^{t-1}\sum_{1\leq j_1<\dots<j_r\leq t}\varepsilon_{j_1}\dots\varepsilon_{j_r}\prod_{\substack{1\leq s\leq t\\ s\notin\{j_1,\dots,j_r\}}}e_{i_s}\\
&= \frac{|F|}{2^t}+\frac{1}{2^t}\sum_{r=1}^{t}\sum_{1\leq j_1<\dots<j_r\leq t}\varepsilon_{j_1}\dots\varepsilon_{j_r}\sum_{E(f)\in F}\prod_{\substack{1\leq s\leq t\\ s\in\{j_1,\dots,j_r\}}}e_{i_s},
\end{aligned}$$

ahol $E(f)=(e_1,\dots,e_{q-1})$. Ezért

$$|G(A)|-\frac{|F|}{2^t}=\frac{1}{2^t}\sum_{u=1}^{t}\sum_{1\leq v_1<\dots<v_u\leq t}\sum_{E(f)\in F}\prod_{z=1}^{u}e_{i_{v_z}}. \tag{3.10}$$

Következésképp elegendő megmutatni, hogy $\sum_{E(f)\in F}\prod_{z=1}^{u}e_{i_{v_z}}=0$. A (3.4) egyenlőség alapján következik, hogy:

$$\sum_{E(f)\in F}\prod_{z=1}^{u}e_{i_{v_z}}=\sum_{E(f)\in F}\prod_{z=1}^{u}\chi(f(\alpha^{i_{v_z}}))=\sum_{E(f)\in F}\chi(f(\alpha^{i_{v_1}})+\dots+f(\alpha^{i_{v_z}})). \tag{3.11}$$

Az $\alpha_i = \sum_{z=1}^{u} \alpha^{(2i+1)i_{v_z}}$ jelöléssel és mivel $f(x) = \sum_{i=0}^{d} a_i x^{2i+1}$:

$$\sum_{E(f)\in F} \chi(f(\alpha^{i_{v_1}}) + \ldots + f(\alpha^{i_{v_z}})) = \sum_{E(f)\in F} \chi(\sum_{i=0}^{d} a_i \sum_{z=1}^{u} \alpha^{(2i+1)i_{v_z}})$$

$$= \sum_{E(f)\in F} \chi(\sum_{i=0}^{d} a_i\alpha_i) = \sum_{\substack{(a_0,\ldots,a_d)\in\mathbb{F}_q^d \\ a_i\neq 0}} \chi(\sum_{i=0}^{d} a_i\alpha_i) \tag{3.12}$$

$$= \left( \sum_{(a_0,\ldots,a_d)\in\mathbb{F}_q^d} \chi(\sum_{i=0}^{d} a_i\alpha_i) - \sum_{j=0}^{d} \sum_{\substack{a_i\in\mathbb{F}_q \\ i\neq j}} \chi( \sum_{\substack{0\leq i\leq d \\ i\neq j}} a_i\alpha_i) \right).$$

Ha $f(x_0, \ldots, x_d) = \sum_{i=0}^{d} x_i\alpha_i$ és minden

$$f_j(x_0, \ldots, x_d) = \sum_{\substack{0\leq i\leq d \\ i\neq j}} x_i\alpha_i, \quad 0 \leq j \leq d$$

nem nulla polinom, akkor a 3.12 Lemma alapján permutációs polinomok, ezért (3.10), (3.11), (3.12) egyenletek és a 3.10 Lemma alapján

$$|G(A)| = \frac{|F|}{2^t}$$

következik.

Ha $f$ vagy bármely $f_j$ nulla polinom, akkor létezik $k, l \in \mathbb{Z}$ úgy, hogy $\alpha_i = 0$ minden $i \in I = \{k, \ldots, k + l - 1\}$ értékre, ahol $l \geq \lfloor \frac{d+1}{2} \rfloor$. Mivel $\alpha_i = \sum_{z=1}^{u} \alpha^{(2i+1)i_{v_z}}$, ez azt jelenti, hogy $\beta_i = \alpha^{2i+1}$ gyöke a $g(x) = \sum_{z=1}^{u} x^{i_{v_z}}$, $g(x) \in \mathbb{F}_2[x]$ polinomnak minden $i \in I$ értékre. Következésképp, ha $m_i(x)$ jelöli a $\beta_i$ minimálpolinomját $\mathbb{F}_2$ felett, akkor minden $\beta_i$

osztja a $g(x)$ polinomot. Mivel a minimálpolinomok irreducibilisek és az egyértelmű faktorizáció miatt, $g(x)$ kódszó a $h(x) = \prod_{i=k}^{k+l} m_i(x)$ szorzat által generált $C$ BCH kódban. A BCH korlátnak köszönhetően (3.9 Lemma) a $C$ kód minimális távolsága legalább $l+1$. Ebből az következik, hogy $g(x)$ súlya legalább $\lfloor \frac{d+1}{2} \rfloor + 1$. Ez ellentmond a feltételezésnek, miszerint az $A$ specifikációnak legfeljebb $\lfloor \frac{d+1}{2} \rfloor$ tagja van. □

**3.2. Folyomány.** *Az $F$ család $f$ bonyolultsága legalább $\lfloor \frac{d+1}{2} \rfloor$.*

## 3.4. Lineárisan visszacsatolt léptetőregiszterek

A [13] cikkben a szerzők az általuk vizsgált generátor lineáris bonyolultság profiljára vonatkozó eredményeiket a korrelációs mértékek segítségével nyerték. Az általuk bejárt gondolatmenet alkalmazásához szükséges, hogy a magasabb rendű korrelációs mértékek is jók legyenek. Ez a bizonyítási út tehát jelen generátor esetében nem alkalmazható.

A jelen szakasz fő eredménye, hogy a lineáris bonyolultság meghatározásán túl, a szóban forgó generátor kapcsolatára is fényt derít a lineáris rekurzív sorozatokkal. Ezáltal a Sárközy és Mauduit által megalapozott álvéletlen mértékeit is meghatározhatjuk a lineáris rekurzív sorozatok egy speciális osztályának. Ugyanakkor ez a szoros kapcsolat azt is lehetővé teszi, hogy a jelen fejezet témájául szolgáló generátort lineárisan visszacsatolt léptetőregiszterekkel implementáljuk, ami nagyon gyors működési sebességet tesz lehetővé. Mindent összevetve, a generátor, habár az alacsony lineáris komplexitása miatt kriptográfiai felhasználásra nem alkalmas, nagyon jó statisztikai tulajdonságokkal és álvéletlen mértékekkel rendelkezik, és ugyanakkor nagyon gyors hardveres implementációja is létezik.

Mivel a szóban forgó generátor magasabb rendű korrelációs mértékei nagyon nagyok is lehetnek, ezért a lineáris bonyolultság vizsgálata a korrelációs mértékre támaszkodva nem lehetséges. Ezért a bizonyításhoz szükségünk lesz a lineárisan visszacsatolt léptetőregiszterek és a rekurzív sorozatok fogalmára, továbbá néhány rájuk vonatkozó eredményre:

**3.13. Definíció.** *Legyen $k$ pozitív egész, és legyenek $a, a_0, \ldots, a_{k-1}$ az $\mathbb{F}_q$adott elemei. Az $\mathbb{F}_q$-beli elemek egy $s_0, s_1, \ldots$ sorozatát, amely kielégíti az*

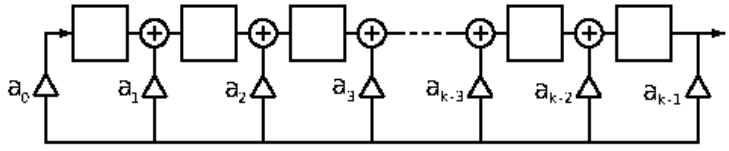
$$s_{n+k} = a_{k-1}s_{n+k-1} + a_{k-2}s_{n+k-2} + \ldots + a_0 s_n + a \qquad (3.13)$$

*egyenlőséget minden $n = 0, 1, \ldots$ esetén, ($k$-ad rendű) lineáris rekurzív sorozatnak nevezzük. A lineáris rekurzív sorozatot homogénnek mondjuk, ha $a = 0$ és inhomogénnek egyébként.*

Lineárisan rekurzív sorozatok generálását hardveresen lineárisan visszacsatolt léptetőregiszterekkel is lehet implementálni. A lineárisan visszacsatolt léptetőregiszterek négyféle építőelemből állnak:

- konstans összeadó
- konstans szorzó
- összeadó
- késleltető

A lineárisan visszacsatolt léptetőregiszterek véges sok ilyen elem egymáshoz illesztésével állnak elő oly módon, hogy zárt hurkot alkossanak. A 3.1 ábrán látható egy lineárisan visszacsatolt léptetőregiszter, amely a 3.13 definícióban megadott homogén lineáris rekurzív sorozatot generálja. A téglalapok jelölik a késleltető elemeket, a háromszögek a konstansszorzókat, a körök pedig az összeadókat. Inhomogén sorozat esetén konstans összeadó elemmel történik meg a konstans tag hozzáadása.




3.1. ábra. Lineárisan visszacsatolt léptetőregiszter (LFSR)

**3.14. Definíció.** *Legyen $s_0, s_1, \ldots$ egy $k$-ad rendű homogén lineáris rekurzív sorozat, amely kielégíti a*

$$s_{n+k} = a_{k-1}s_{n+k-1} + a_{k-2}s_{n+k-2} + \ldots + a_0 s_n$$

*lineáris rekurziót minden $n = 0, 1, \ldots$ értékre, ahol $a_j \in \mathbb{F}_q$ minden $q \leq j \leq k-1$ esetén. Ekkor az*

$$f(x) = x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \ldots - a_0 \in \mathbb{F}_q[x] \tag{3.14}$$

*polinomot a lineáris rekurzív sorozat karakterisztikus polinomjának nevezzük.*

**3.15. Definíció.** *Legyen $s_0, s_1, \ldots$ egy $k$-ad rendű homogén lineáris rekurzív sorozat $\mathbb{F}_q$ felett. Ekkor egyértelműen létezik olyan $m(x) \in \mathbb{F}_q[x]$ egy főegyüttható́jú polinom, amelyre teljesül, hogy egy pozitív fokszámú $f(x) \in \mathbb{F}_q[x]$ egy főegyüttható́jú polinom akkor és csakis akkor lesz a sorozat karakterisztikus polinomja, ha $m(x)$ osztja $f(x)$ -et.*

**3.13. Lemma.** *Legyen $s_0, s_1, \ldots$ egy $k$-ad rendű homogén lineáris rekurzív sorozat $K = \mathbb{F}_q$ felett, amelynek $f(x)$ karakterisztikus polinomja irreducibilis $K$ felett. Legyen $\alpha$ az $f(x)$ gyöke a $F = \mathbb{F}_{q^k}$ testben. Ekkor egyértelműen létezik olyan $\theta \in F$, hogy*

$$s_n = Tr_{F/K}(\theta\alpha^n) \text{ for } n = 0, 1, \ldots.$$

*Bizonyítás.* Ez a 8.24. tétel a [48] könyvben. □

**3.14. Lemma.** *Legyen az $f(x)$ egy $k$ fokú irreducibilis polinom $K = \mathbb{F}_q$ felett, $\alpha$ pedig egy gyöke a $F = \mathbb{F}_{q^k}$ testben és $\theta \in F$. Ekkor egyértelműen létezik olyan $s_0, s_1, \ldots$ homogén lineáris rekurzív sorozat $F$ felett, amelynek $f(x)$ karakterisztikus polinomja, és*

$$s_n = Tr_{F/K}(\theta\alpha^n) \text{ for } n = 0, 1, \ldots.$$

*Bizonyítás.* Ez a 3.13 Lemmából következik és abból a tényből, hogy az $F$ testnek pontosan $q^k$ eleme van és éppen ennyi homogén lineáris rekurzív sorozat létezik $K$ felett, amelynek $f(x)$ a karakterisztikus polinomja. □

**3.15. Lemma.** *Legyen $f(x) \in \mathbb{F}_q[x]$ főegyütthatója 1 és legyen irreducibilis $\mathbb{F}_q$ felett, legyen továbbá $s_0, s_1, \ldots$ olyan homogén lineáris rekurzív sorozat $\mathbb{F}_q$ felett, aminek nem minden tagja nulla. Ha a sorozatnak $f(x)$ karakterisztikus polinomja, akkor a minimálpolinomja éppen $f(x)$.*

*Bizonyítás.* Ez a 8.50. tétel a [48] könyvben. □

Definiáljuk ezek után a sorozatok tagonkénti összeadását:

**3.16. Lemma.** *Minden $i = 1, 2, \ldots, h$ esetén, legyen $\sigma_i$ homogén lineáris rekurzív sorozat $\mathbb{F}_q$ felett, aminek a minimálpolinomja $m_i \in \mathbb{F}_q[x]$. Ha a $m_1(x), \ldots, m_h(x)$ polinomok páronként relatív prímek, akkor a $\sigma_1 + \ldots + \sigma_h$ összeg minimálpolinomja $m_1(x) \ldots m_h(x)$.*

*Bizonyítás.* Ez a 8.57. tétel a [48] könyvben. □

A korábbiakban a vizsgált bináris sorozatot $\{-1, 1\}^N$ típusúnak definiáltuk Ahhoz, hogy a lineáris rekurzív sorozatokkal összefüggésben tudjuk vizsgálni, definiáljuk a következő sorozatot.

**3.16. Definíció.** *Legyen $f(x)$ továbbra is olyan, hogy megfelel a 3.2 Tétel feltételeinek. Legyen ekkor a bináris álvéletlen sorozat:*

$$e_n = \begin{cases} 1, & \text{ha } \chi(f(\alpha^n)) = -1, \\ 0, & \text{egyébként.} \end{cases} \tag{3.15}$$

**3.4. Megjegyzés.** *Könnyen látható, hogy ez a sorozat éppen*

$$e_n = Tr(f(\alpha^n)),$$

*ahol $Tr(x)$ az abszolút nyom függvényt jelöli.*

**3.17. Definíció.** *Egy sorozat lineáris bonyolultságán a legrövidebb olyan lineárisan visszacsatolt léptetőregiszter bitekben kifejezett hosszát értjük, amely a sorozatot generálni tudja.*

**3.6. Tétel.** *A 3.16 definícióban megadott bináris sorozat lineáris bonyolultsága* $\frac{k(d+1)}{2}$.

*Bizonyítás.* Legyen $f(x) = \sum_{i=0}^{l} \theta_i x^{2i+1}$, ahol $l = \frac{d-1}{2}$. Ekkor a $\alpha_i = \alpha^{2i+1}$, $i = 0, \ldots, l$ jelöléssel kapjuk, hogy

$$e_n = Tr_{F/K}(f(\alpha^n)) = \sum_{i=0}^{l} Tr_{F/K}(\theta_i \alpha^{n(2i+1)}) = \sum_{i=0}^{l} Tr_{F/K}(\theta_i \alpha_i^n).$$

Az $\alpha_i$ definiálópolinomját $\mathbb{F}_2$ felett jelölje $m_i(x)$. A 3.14 Lemma alkalmazásával következik, hogy

$$(e_n) = \sigma_0 + \ldots + \sigma_l,$$

ahol $\sigma_i$ homogén lineáris rekurzív sorozat $\mathbb{F}_2$ felett $m_i(x)$ karakterisztikus polinommal. A 3.15 Lemma szerint az $m_i(x)$ polinom nem csak egy karakterisztikus polinom, de $\sigma_i$ minimálpolinomja is. Ekkor a 3.16 Lemma alapján $(e_n)$ is homogén lineáris rekurzív sorozat $\mathbb{F}_2$ felett $M(x) = \prod_{i=0}^{l} m_i(x)$ minimálpolinommal. Mivel mindegyik $m_i(x)$ foka $k$, az $M(x)$ fokszáma és az $(e_n) = \sigma_0 + \ldots + \sigma_l$ sorozat rendje $\frac{k(d+1)}{2}$. □

# 4. fejezet

# Hash függvények

A fejezetben egy új hash függvény konstrukcióról lesz szó. Az első szakasz a hash függvényekről szól általánosságban. A második szakasz a konstrukció Codefish néven implementált elődjét ismerteti ([10]), a harmadik pedig annak kriptanalízisével foglalkozik. Az új konstrukció és a vele kapcsolatos elméleti eredmények a negyedik szakaszban kaptak helyet. A konstrukció kiküszöböli elődje hibáit ([6]) és az operandusok méretében is előrelépést jelent. Elméleti megfontolások alapozzák meg a függvény előkép-ellenállóságát és egy jó statisztikai tulajdonsága is bizonyítást nyer. Ezen hash függvény előkép-ellenállására vonatkozó megállapítások Bérczes Attilával és Pethő Attilával közös eredményeink és a [7] cikkben kerültek publikálásra. A lavinahatásra és a hozzá kapcsolódó asszimptotikus állításra vonatkozó tétel saját eredmény és eddig még nem lett publikálva.

## 4.1. Bevezetés

A kriptográfiai hash függvények elsődleges feladata az adatintegritás biztosítása. Az elv az, hogy valamely adatról lenyomatot készítünk a hash függvénnyel. Ebben a felállásban az adat tetszőleges hosszúságú bitsztringet jelent, tipikusan hosszabbat, mint a lenyomat. A lenyomat valamilyen fix hosszúságú ellenőrző összeg, ez a hossz a gyakorlatban alkalmazott

kriptográfiai hash függvényeknél rendszerint legalább 160 bit. Ezek után, bármikor meggyőződhetünk az eredeti adat változatlanságáról, azzal, hogy ismét elkészítjük az adat lenyomatát és amennyiben nem egyezik a régivel, azonnal tudjuk, hogy változás történt.

A kriptográfiai hash függvények fontos szerepet játszanak a digitális aláírás sémákban. A digitális aláíró algoritmusok rendszerint első lépésben lenyomatot készítenek az aláírandó adatról, és csak a lenyomatot írják alá.

Szintén gyakori, hogy a jelszavas azonosítást használó rendszerek nem a jelszavakat tárolják, hanem a jelszavak lenyomatait és a felhasználók beléptetése, azaz a jelszavak ellenőrzése is a lenyomatok alapján történik. Például a Unix alapú operációs rendszerek is ezt az eljárást követik: a jelszavak lenyomatai csak a gyökérfelhasználó által hozzáférhető /etc/shadow (BSD esetén /etc/master.passwd) fájlban vannak.

### 4.1.1. Egyirányú függvények

A kriptográfiai hash függvények szoros kapcsolatban állnak az egyirányú függvény fogalmával. Szabadon fogalmazva az egyirányú függvény egy olyan függvény, amelyet *könnyű* kiszámítani, de az inverzeinek a kiszámítása már *nehéz* feladat ([33] 33. oldal 2.2.1 Definíció):

**4.1. Definíció** (Egyirányú függvény)**.** *Egy $f : \{0,1\}^* \to \{0,1\}^*$ függvény egyirányú, ha a következő két feltétel teljesül:*

1. Könnyű számítani: *Létezik olyan polinomidejű $\mathcal{A}$ algoritmus, hogy $x$ bemenet esetén a kimenete $f(x)$.*

2. Nehéz invertálni: *Minden $\mathcal{A}'$ polinomidejű valószínűségi algoritmusra, minden pozitív $p(\cdot)$ polinomra elegendően nagy $n$ esetén teljesül, hogy*

$$P[\mathcal{A}'(f(U_n), 1^n) \in f^{-1}(f(U_n))] < \frac{1}{p(n)},$$

*ahol $U_n$ egyenletes eloszlású valószínűségi változó, amely a $\{0,1\}^n$ halmazból veszi értékeit.*

Bonyolultságelméleti megfogalmazásban ez alatt azt értjük, hogy a függvény maga legfeljebb polinomidejű algoritmussal számolható, azonban az inverzei csupán legalább exponenciális idejű algoritmussal számíthatóak ki. Ha egy függvény polinomidőben számolható, akkor a **P** osztályba tartozik, az inverzei pedig az **NP** osztályba. Ebben az értelemben tehát csak akkor létezhet egyirányú függvény, ha $\mathbf{P} \neq \mathbf{NP}$.

Mivel nem tudjuk, hogy $\mathbf{P} \neq \mathbf{NP}$ igaz-e, ezért azt sem tudhatjuk, hogy a fenti értelemben létezik-e egyirányú függvény. Ennek ellenére több javaslat is megjelent az irodalomban egyirányú függvényeket illetően: [69] [58] [18] [41] [14]. A [32] cikkben például a szerzők az RSA probléma és a diszkrét logaritmus probléma segítségével konstruálnak egyirányú függvényt.

### 4.1.2. Biztonsági kritériumok

**4.2. Definíció.** *Legyen $\mathcal{X}$ a lehetséges üzenetek halmaza, $\mathcal{Y}$ pedig a lehetséges lenyomatok véges halmaza, és teljesüljön rájuk, hogy $|\mathcal{X}| \geq |\mathcal{Y}|$. Ekkor a $h : \mathcal{X} \to \mathcal{Y}$ függvényt hash függvénynek nevezzük.*

A hash függvényekkel szemben három különböző biztonsági kritériumot szokás megfogalmazni:

1. Őskép-ellenállás
2. 2. őskép-ellenállás
3. Ütközés-ellenállás

Ezek különböző szempontokat és különböző erőt képviselnek. Legyen $y = h(x)$, ekkor az az elvárásunk a hash függvénnyel kapcsolatban, hogy érvényes $(x, y)$ pár előállításának az egyetlen módja a $h$ függvény $x$ értékre való alkalmazása legyen.

**4.1. Probléma.** *(Őskép)*
Adott: *a $h : \mathcal{X} \to \mathcal{Y}$ hash függvény és egy $y \in \mathcal{Y}$ elem.*
Feladat: *olyan $x \in \mathcal{X}$ értéket találni, hogy $h(x) = y$.*

Egy $h$ hash függvény őskép-ellenálló, ha a hozzá kapcsolódó 4.1 Probléma nem oldható meg hatékonyan. Az őskép-ellenállás szoros kapcsolatban van az egyirányú függvény fogalmával.

**4.2. Probléma.** *(Második őskép)*
Adott: *a* $h : \mathcal{X} \to \mathcal{Y}$ *hash függvény és egy* $x \in \mathcal{X}$ *elem.*
Feladat: *olyan* $x' \in \mathcal{X}$ *értéket találni, hogy* $x' \neq x$ *és* $h(x) = h(x')$.

Egy $h$ hash függvény második őskép-ellenálló, ha a hozzá kapcsolódó 4.2 Probléma nem oldható meg hatékonyan.

**4.3. Probléma.** *(Ütközés)*
Adott: *a* $h : \mathcal{X} \to \mathcal{Y}$ *hash függvény.*
Feladat: *olyan* $x, x' \in \mathcal{X}$ *értékeket találni, hogy* $h(x) = h(x')$

Egy $h$ hash függvény ütközés-ellenálló, ha a hozzá kapcsolódó 4.3 Probléma nem oldható meg hatékonyan.

## 4.2. Codefish

A vizsgálatunk tárgyát képező kriptográfiai hash függvény közvetlen elődje a [10] cikkben ismertetett hash függvény, amelyet a Kripto Kft. Codefish néven hozott kereskedelmi forgalomba.

A Codefish elméleti hátterének vizsgálatához szükségünk van a normaformák egy általánosabb változatának a megadására.

**4.3. Definíció.** *Legyen* $P(X) \in \mathbb{Z}[X]$ *egy fix polinom, amelynek a főegyütthatója egy és a fokszáma* $n \geq 3$ *és nincs többszörös gyöke. Legyenek* $\alpha_1, \ldots, \alpha_n$ *a* $P$ *gyökei és legyen*

$$L_i(\underline{X}) = \sum_{j=1}^{m} \alpha_i^{j-1} X_j \text{ ahol } i = 1, \ldots, n \text{ és } m \leq n.$$

*Definiáljuk a* $P$ *polinomhoz tartozó normaformát a következőképp:*

$$\mathcal{N}_P(\underline{X}) = \prod_{i=1}^{n} L_i(\underline{X}).$$

**4.1. Megjegyzés.** *Az $\mathcal{N}_P(\underline{X})$ a normaforma koncepciójának továbbvitele és egy felbontható forma. A $\mathcal{N}_P(\underline{X})$ polinom homogén, fokszáma $n$ és együtthatói egészek.*

A fenti normaforma fogalom segítségével megadhatjuk a következő leképezést:

**4.4. Definíció.** *Definiáljuk az $\mathcal{N}_P : \mathbb{Z}^m \to \mathbb{Z}$ leképezést a következőképp:*

$$\mathcal{N}_P : (x_1, \ldots, x_m) \to \mathcal{N}_P(x_1, \ldots, x_m). \tag{4.1}$$

A gyakorlati alkalmazhatóság szempontjából elemi fontosságú, hogy a függvény polinomidőben számítható legyen. A [9] cikkben a szerzők az $\mathcal{N}_P(\underline{X})$ számítási bonyolultságát vizsgálták irreducibilis $P$ esetén. Bizonyítást nyert, hogy mátrix reprezentáció esetén $\mathcal{O}(n^7 + n^6 \log \mathbb{X} + n^2 \log^{2/3} \mathbb{X})$, ahol $n$ a $P$ polinom fokszámát jelöli, $\mathbb{X} = max\{|x_1|, \ldots, |x_m|, 1\}$ és a $\mathcal{O}$ jelölésben szereplő konstans csakis a $P$ együtthatóinak abszolút értékének a maximumától függ.

Gyakorlati szempontból praktikusabb véges struktúrák felett dolgozni, ezért definiáljuk a fenti függvényt véges esetre:

**4.5. Definíció.** *Legyen $s$ egész és definiáljuk a $\mathcal{N}_{P,s} : \mathbb{Z}_s^m \to \mathbb{Z}_s$ leképezést a következőképp:*

$$\mathcal{N}_P : (x_1, \ldots, x_m) \to \mathcal{N}_P(x_1, \ldots, x_m) \mod s. \tag{4.2}$$

Az $s$ modulus szerinti számolás tovább gyorsítja a leképezés kiértékelését:

**4.1. Lemma.** *Az $\mathcal{N}_{P,s}(\underline{x})$ kiszámításának bonyolultsága a [9] cikkben a 2. tételben leírt algoritmus segítségével $\mathcal{O}(n^5 \log^2 s)$, ahol az $\mathcal{O}$ jelölés jelentette konstans csakis a $P(X)$ polinomtól függ.*

*Bizonyítás.* Ez az 1. Tétel a [10] cikkben. □

Jelenleg nincs ismert algoritmus általános normaforma egyenletek összes megoldásának meghatározására, azaz az $\mathcal{N}_{P,s}$ leképezés invertálására. Ezt fogalmazza meg az erős moduláris normaforma feltevés:

**4.6. Definíció** (Erős Moduláris Normaforma Feltevés)**.** *Erős Moduláris Normaforma Feltevésnek nevezzük azt a feltételezést, miszerint a 4.5 definícióban meghatározott $\mathcal{N}_{P,s}$ leképezés olyan, hogy minden $Q$ polinom és bármely $\mathcal{A}$ polinomidejű valószínűségi algoritmus esetén elegendően nagy $s$ egészekre:*

$$P[\mathcal{A}(s, \mathcal{N}_{P,s}(x_1, \ldots, x_m)) = (x_1, \ldots, x_m)] < \frac{1}{Q(s)},$$

*ahol $x_i \in \mathbb{Z}_s$ és a valószínűséget minden lehetséges $x_i$ érték és $\mathcal{A}$ algoritmusbeli érmedobás felett értjük.*

Ezen feltevés mellett a $\mathcal{N}_{P,s}$ leképezés egyirányúsága és a vele összefüggő őskép-ellenállóság triviálisan teljesül.

Kézenfekvő választás volna az $s$ értékéül valamely kellően nagy prímszámot választani, azonban ebben az esetben nem teljesül az Erős Moduláris Normaforma Feltevés:

**4.2. Lemma.** *Legyen $s$ prím, $P \in \mathbb{Z}[X]$ és $b \in \mathbb{Z}_s$. Ekkor létezik olyan polinomidejű valószínűségi algoritmus, amely $\mathcal{N}_{P,s}(\underline{x}) = b$ ismeretében kiszámolja az $\underline{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_s^m$ előképet.*

*Bizonyítás.* Ez az 1. javaslat a [10] cikkben. □

Ez a feltétel nem elegendő az ütközés-ellenálláshoz. A következő eredmény azonban a leképezés ütközés-ellenállósága felé mutat abban az értelemben, hogy a megfelelően megválasztott paraméterek mellett ellenáll a születésnap paradoxonra épülő támadásoknak.

**4.3. Lemma.** *Legyen $P(X) \in \mathbb{Z}[X]$ egy főegyütthatójú legalább harmadfokú polinom, amelynek nincsenek többszörös gyökei. Legyenek $p$ és $q$ prímek úgy, hogy $q > p > q/2$ és legyen $s = pq$. Tegyük fel, hogy $gcd(m, \varphi(s)) = 1$. Jelölje $N(P, b, s)$ az $\mathcal{N}_P(x_1, \ldots, x_m) \equiv b \mod s$ kongruencia megoldásainak a számát.*

1. *Ha $gcd(b, s) = 1$, akkor*

$$|N(P, b, s) - s^{m-1}| < c_1(P) s^{m-1-\frac{1}{4}},$$

*2. egyébként pedig*

$$N(P, b, s) < c_2(P)s^{m-1}.$$

*Bizonyítás.* Ez az 5. tétel a [10] cikkben. □

## 4.3. A Codefish kriptoanalízise

A Kripto Kft. által implementált Codefish nevű kriptográfiai hash függvény egy iteratív hash függvény, és mint ilyen, két részből áll:

1. Egy tömörítő függvényből és
2. Egy iteráló módszerből

A tömörítő függvény rész, az előző szakaszban ismertetett leképezésen alapszik. Az implementációban a [9] -ban ismertetett mátrixreprezentáción alapuló algoritmust használták. Ennek megfelelően a tömörítőfüggvény használata az

$$\mathcal{N}(X_1, \ldots, X_n) = \det \begin{pmatrix} X_1 & X_2 & \ldots & X_n \\ X_n & X_1 & \ldots & X_{n-1} \\ \ldots & \ldots & \ldots & \ldots \\ X_2 & X_3 & \ldots & X_1 \end{pmatrix}$$

cirkuláris mátrix determinánsának kiszámítását jelenti.

A Kripto Kft. implementációjában egy további $m$ paraméter járul a függvényhez, aminek segítségével a $\mathcal{N}$ kevesebb szót képes fogadni a bemeneten és ezáltal még gyorsabban számolható a leképezés:

$$\mathcal{N}(X_1, \ldots, X_m) = \det \begin{pmatrix} X_1 & X_2 & \ldots & X_m & 0 & \ldots & 0 \\ 0 & X_1 & \ldots & X_{m-1} & X_m & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ X_2 & X_3 & \ldots & 0 & 0 & \ldots & X_1 \end{pmatrix}.$$

Másrészt a Codefish egy meglehetősen szokatlan iterációs sémát használ. Az $X_1, \ldots, X_l$ $l > n$ bemenet hasheléséhez először kiszámoljuk a $H_1 = \mathcal{N}(X_1, \ldots, X_n)$ hash értéket, aztán a továbbiakban a

$$H_{i+1} = \mathcal{N}(H_i, X_{n+i(n-1)}, \ldots, X_{n+(i+1)(n-1)})$$

rekurziós formulát használva számítjuk a hash értéket.

A [6] cikkben a szerző gyakorlati támadásokat ad a Codefish ellen a 2. előkép- és az ütközés-ellenállóság tulajdonságra vonatkozóan. Ezek a támadások nem mondanak ellent a fent vázolt elméleti eredményeknek és a gyakorlati megvalósíthatóságuk is elsősorban az implementációs döntéseknek köszönhetőek.

Nevezetesen, hogy a Codefish egy nem konvencionális iterációs sémát használ, továbbá, hogy a sebesség növelésének érdekében a konstrukciót is leegyszerűsítették. Szerepet játszik a támadások sikerében az is, hogy a nem egész blokkokból álló bemeneteket egyszerűen nullákkal tölti fel a program.

Az egyetlen elméleti szinten is megjelenő gyengeség, a tömörítő függvény homomorf tulajdonsága. Nevezetesen, hogy mivel $A$ és $B$ ugyanazon kommutatív gyűrű feletti $n \times n$ mátrixok, ezért $\det(A) \times \det(B) = \det(AB)$ és két cirkuláris mátrix szorzata is cirkuláris:

$$\mathcal{N}(X_1, \ldots, X_n) \times \mathcal{N}(Y_1, \ldots, Y_n) = \mathcal{N}(Z_1, \ldots, Z_n),$$

ahol $Z_i = \sum_{j=1}^{n} X_j Y_{n-j+1+i},\ i = 1, \ldots, n$, a $X_1, \ldots, X_n$ és $Y_1, \ldots, Y_n$ pedig tetszőleges bemeneti blokkok. Ez lehetővé teszi két lenyomat ismeretében egy harmadik kiszámítását a támadó számára, anélkül, hogy az $\mathcal{N}$ leképezést kiértékelné.

## 4.4. UDHash

A Codefish elméleti alapjául szolgáló elvet a [8] cikkben indexformákra is alkalmazták. A jelen szakasz fő témájául szolgáló hash függvény a [7] cikkben került publikálásra, és ugyanezt az elvet használja, viszont több téren is előrelépést jelent.

Egyrészt kiküszöböli az előző szakaszban említett nemkívánatos homomorf tulajdonságot, másrészt páros karakterisztikájú véges testek feletti műveleteken alapszik.

A Codefishnél $s$ két nagy prím szorzata kellett, hogy legyen. A kriptográfiai biztonsághoz ez jellemzően 1024 bites vagy még hosszabb modulust

jelent. A kettő karakterisztikájú $\mathbb{F}_q$ test alkalmazása esetén a biztonsághoz elegendő, ha $q$ jelentősen rövidebb, például a hagyományos, blokk kódolókon alapuló hash függvényekéhez hasonló méretű (például 256 bit). A kettő karakterisztikájú véges testek alkalmazásának másik előnye, hogy hardver implementáció esetén nagyobb műveleti sebesség elérését teszi lehetővé.

**4.7. Definíció** (UDHash)**.** *Legyen az* $f(\underline{X}) \in \mathbb{F}_q[X_1, \ldots, X_m]$ *polinom a következő formájú:*

$$f(\underline{X}) = b(X_1, \ldots, X_m) + a(X_1, \ldots, X_m),$$

*ahol* $a(\underline{X})$ *és* $b(\underline{X})$ *homogén polinomok, amelyeknek a fokszámaira teljesül, hogy* $k = \deg a(\underline{X}) < \deg b(\underline{X}) = n$ *és* $\deg_{X_i} b(\underline{X}) = n$ *minden* $1 \leq i \leq m$ *esetén. Tegyük fel továbbá, hogy léteznek olyan* $1 \leq j_1 < j_2 \leq n$ *indexek, hogy a*

$$b_o(X_{j_1}, X_{j_2}) = b(0, \ldots, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0)$$

*bináris formának nincs többszörös gyöke.*

A fenti definíciónak a hash függvények egy széles családja felel meg. Ezen hash függvények szempontjából az előkép-ellenállóság a következőt jelenti:

**4.8. Definíció.** *Legyen az* $f(\underline{X}) \in \mathbb{F}_q[X_1, \ldots, X_m]$ *polinom olyan, hogy megfelel a 4.7 Definíciónak. Az* $f(\underline{X})$ *polinomfüggvény előkép-ellenálló, ha minden* $Q$ *polinom,* $\gamma \in \mathbb{F}_q$*, és bármely* $\mathcal{A}$ *polinomidejű valószínűségi algoritmus esetén kellően nagy* $q = 2^k$ *egészekre teljesül, hogy*

$$P[\mathcal{A}(q, \gamma = f(\underline{x})) = \underline{x}] < \frac{1}{Q(q)},$$

*ahol* $\underline{x} \in \mathbb{F}_q^n$ *és a valószínűséget minden lehetséges* $\underline{x}$ *érték és* $\mathcal{A}$ *algoritmusbeli érmedobás felett értjük.*

Több algoritmus is ismert a véges testek feletti egyenletek megoldására. Ilyen a Berlekamp féle [11] és az LLL algoritmus [47]. Ezek a legjobb ismert módszerek a probléma megoldására és mindkettő exponenciális $q$-ban, azaz a szóban forgó test méretében.

Shparlinski [67] azzal érvel a diszkrét logaritmus probléma nehézsége mellett, hogy a diszkrét logaritmus függvény nem reprezentálható alacsony fokszámú polinommal.

Habár az LLL algoritmus polinomiális a polinom fokszámában, ha az a test méretéhez mérhető, az algoritmus nem jelent javulást.

Mint ahogy a Codefish esetében, így itt is az előkép-ellenállóság a feltételezésből triviálisan következik, azonban a 2. előkép-, illetve az ütközés-ellenállóságot nem vonja maga után.

Az UDHash esetében is elmondhatjuk, hogy ugyan az ütközés-ellenállóságot nem tudjuk bizonyítani, de kellően nagy $q$ esetén ellenáll a születésnap típusú támadásoknak. Az eredmény bizonyításához szükségünk lesz a következő eredményekre:

A bizonyítás részben Cafure és Matera [15] (vö. [46, 64]) $\mathbb{F}_q$ azon pontjaira vonatkozó eredményén alapszik, amelyek egy $\mathbb{F}_q$ felett definiált hiperfelületen fekszenek.

**4.4. Lemma.** *Egy abszolút irreducibilis $\delta$ fokú $\mathbb{A}^n$ feletti $H$ $\mathbb{F}_q$-hiperfelület esetén a következő becslés érvényes:*

$$\left||H \cap \mathbb{F}_q^n| - q^{n-1}\right| \leq (\delta-1)(\delta-2)q^{n-3/2} + 5\delta^{13/3}q^{n-2}.$$

A tételben $\mathbb{A}^n$ jelöli az $\mathbb{F}_q$ feletti $n$ dimenziós affin teret. Ha a $q$ elegendően nagy, az állítás sokkal jobb maradéktaggal is igazolható [15].

**4.5. Lemma.** *Legyen $q > 15\delta^{13/3}$ és legyen $H \subseteq \mathbb{A}^n$ abszolút irreducibilis $\delta$ fokú $\mathbb{F}_q$ -hiperfelület. Ekkor a következő egyenlőtlenség teljesül*

$$\left||H \cap \mathbb{F}_q^n| - q^{n-1}\right| \leq (\delta-1)(\delta-2)q^{n-3/2} + (5\delta^2+\delta+1)q^{n-2}.$$

**4.6. Lemma.** *Legyen $K$ egy tetszőleges test, és legyen $\overline{K}$ a $K$ test algebrai lezárása. Legyen $n \geq 4$ egész, és legyen*

$$G(X,Y) = Y^n + A(X)Y^{n-1} + B(X) \in K[X,Y]$$

*egy polinom a következő tulajdonságokkal: $A(X), B(X) \in K[X]$, $B(X)$-nek nincs többszörös gyöke és $\deg A(X) \neq \deg B(X) \geq 1$. Ekkor $G(X,Y)$ irreducibilis $\overline{K}$ felett, azaz abszolút irreducibilis.*

*Bizonyítás.* Tegyük fel indirekten, hogy $G(X,Y)$ reducibilis, azaz, hogy $G(X,Y) = U(X,Y)V(X,Y)$, ahol

$$U(X,Y) = Y^k + a_{k-1}(X)Y^{k-1} + \cdots + a_1(X)Y + a_0(X) \in \overline{K}[X,Y],$$
$$V(X,Y) = Y^{n-k} + b_{n-k-1}(X)Y^{n-k-1} + \cdots + b_1(X)Y + b_0(X) \in \overline{K}[X,Y],$$

és $1 \leq k \leq n-1$, $a_i(X), b_j(X) \in \overline{K}[X]$ minden $i, j \in Z_{\geq 0}$ esetén, $a_k(X) = 1$, $b_{n-k}(X) = 1$, továbbá $a_i(X) = 0$, minden $i > k$-ra és $b_j(X) = 0$ minden $j > n-k$-ra konstans polinomok.
*I. Eset:* Először tegyük fel, hogy $min(k, n-k) \geq 2$. Ekkor

$$G(X,Y) = U(X,Y)V(X,Y) = \sum_{i=0}^{n} c_i(X)Y^i, \tag{4.3}$$

és

$$c_i(X) = \sum_{j=0}^{i} a_j(X)b_{i-j}(X). \tag{4.4}$$

Mivel $\deg B(X) \geq 1$, az általánosság megsértése nélkül feltételezhetjük, hogy $\deg a_0(X) \geq 1$. Ekkor létezik olyan $\alpha \in \overline{K}$, hogy $a_0(\alpha) = 0$. Mivel $B(X) = a_0(X)b_0(X)$, és $B(X)$ -nek nincs többszörös gyöke, azt kapjuk, hogy $b_0(\alpha) \neq 0$. (4.3) és $G(X,Y) = Y^n + A(X)Y^{n-1} + B(X)$ összehasonlításával azt kapjuk, hogy $c_i(X) = 0$ a konstans 0 polinom minden $i = 1, \ldots, n-2$-re. Tehát $c_i(\alpha) = 0$ minden $i = 1, \ldots, n-2$ -re, ami (4.4)-al együtt arra vezet, hogy

$$\sum_{j=0}^{i} a_j(\alpha)b_{i-j}(\alpha) = 0 \qquad \text{for } \; i = 1, \ldots, n-2. \tag{4.5}$$

Ekkor (4.5) teljesül minden $i =$ 1-re, ez azzal együtt, hogy $a_0(\alpha) = 0$ és $b_0(\alpha) \neq 0$ azt bizonyítja, hogy $a_1(\alpha) = 0$. Hasonlóan, (4.5) teljesül minden $i = l$ -re, és ez azzal együtt, hogy $a_0(\alpha) = 0, \ldots, a_{l-1}(\alpha) = 0$ és $b_0(\alpha) \neq 0$ bizonyítja, hogy $a_l(\alpha) = 0$ bármely $l = 1, \ldots, n-2$ -re. Tehát arra jutunk,

hogy $a_i(\alpha) = 0$ minden $i = 0, \ldots, n-2$-re. Mivel $\min(k, n-k) \geq 2$ azt kapjuk, hogy $U(\alpha, Y) = Y^k$ és

$$\begin{aligned} Y^n + A(\alpha)Y^{n-1} + B(\alpha) &= U(\alpha, Y)V(\alpha, Y) \\ &= Y^n + b_{n-k-1}(\alpha)Y^{n-1} + \cdots + b_0(\alpha)Y^k, \end{aligned}$$

és ez nyilvánvalóan ellentmondás.
*II. Eset:* Tegyük fel most, hogy $min(k, n-k) = 1$. Az általánosság veszélyeztetése nélkül feltehetjük, hogy $k = 1$. Ekkor

$$\begin{aligned} &U(X, Y) = Y + a_0(X), \\ &V(X, Y) = Y^{n-1} + b_{n-2}(X)Y^{n-2} + \cdots + b_1(X)Y + b_0(X). \end{aligned} \tag{4.6}$$

Ekkor $Y^n + A(X)Y^{n-1} + B(X) = U(X, Y)V(X, Y)$ figyelembe véve, hogy (4.6) azt kapjuk, hogy $B(X) = a_0(X)b_0(X)$, $a_0(X)b_l(X) - b_{l-1}(X) = 0$ minden $l = 1, \ldots, n-2$-re, és $a_0(X) + b_{n-2}(X) = A(X)$ esetén. Az első két egyenlőségből következik, hogy $B(x)$ osztható $a_0(X)^2$-tel, ebből és abból a feltételből, hogy $B(X)$-nek nincs többszörös gyöke következik, hogy $a_0(X) = a$ konstans.

Ekkor a fenti egyenlőségek azt jelentik, hogy $b_{n-k-2}(X) = (-a)^k b_{n-2}(X)$ minden $k = 1, \ldots, n-2$-re. Ebből következik, hogy

$$\begin{aligned} Y^n + A(X)Y^{n-1} + B(X) &= U(X, Y)V(X, Y) \\ &= Y^n + (a + b_{n-2}(X))Y^{n-1} + a(-a)^{n-2}b_{n-2}(X). \end{aligned}$$

Ez pedig azt jelenti, hogy $\deg A(X) = \deg B(X)$, ami ellentmond a Lemma feltételeinek.

Ezzel együtt pedig a 4.6 Lemma bizonyítása teljes. □

**4.7. Lemma.** *Legyen $K$ egy tetszőleges test. Legyen az $f(\underline{X}) \in K[X_1, \ldots, X_m]$ polinom olyan, hogy*

$$f(\underline{X}) = b(X_1, \ldots, X_m) + a(X_1, \ldots, X_m),$$

*ahol $a(\underline{X}), b(\underline{X})$ homogén polinomok, és teljesül rájuk, hogy $k = \deg a(\underline{X}) < \deg b(\underline{X}) = n$, és $\deg_{X_i} b(\underline{X}) = n$ minden $1 \leq i \leq m$ -re. Továbbá tegyük*

*fel, hogy léteznek olyan* $1 \leq j_1 < j_2 \leq m$ *indexek, hogy a*

$$b_0(X_{j_1}, X_{j_2}) = b(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0) \tag{4.7}$$

*bináris formának nincs többszörös gyöke. Ekkor a* $f(\underline{X})+\gamma$ *polinom abszolút irreducibilis minden* $0 \neq \gamma \in K$*-ra.*

*Bizonyítás.* Legyen $g(\underline{X}) = f(\underline{X}) + \gamma$, $f_0(\underline{X}) = b_0(\underline{X}) + a_0(\underline{X})$ és $g_0(\underline{X}) = f_0(\underline{X}) + \gamma$, ahol $a_0(\underline{X}) = a(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0)$.

Tegyük fel indirekten, hogy $g(\underline{X})$ reducibilis, azaz, hogy $g(\underline{X}) = U(\underline{X})V(\underline{X})$, ahol $\deg U(\underline{X}) \geq 1$ és $\deg V(\underline{X}) \geq 1$. Tehát léteznek olyan $i \in \{1, \ldots, m\}$ indexek, hogy $\deg_{X_i} U(\underline{X}) \geq 1$. Ekkor felhasználva, hogy $\deg_{X_j} g(\underline{X}) = n$ minden $j \in \{1, \ldots, m\}$-re, látható, hogy $\deg_{X_j} V(\underline{X}) < n$ és ezért $\deg_{X_j} U(\underline{X}) > 0$ minden $j \in \{1, \ldots, m\}$ -re. Hasonlóan, mivel $\deg_{X_j} U(\underline{X}) < n$ teljesül $\deg_{X_j} V(\underline{X}) > 0$ következik minden $j \in \{1, \ldots, m\}$ -re. Mindent összevetve ez azt jelenti, hogy

$$1 \leq \deg_{X_j} U(\underline{X}) \leq n-1 \quad \text{és} \quad 1 \leq \deg_{X_j} V(\underline{X}) \leq n-1 \text{ minden } j \in \{1, \ldots, m\}\text{-re.} \tag{4.8}$$

Most legyen

$$U_0(X_{j_1}, X_{j_2}) = U(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0)$$

és

$$V_0(X_{j_1}, X_{j_2}) = V(0, \ldots, 0, X_{j_1}, 0, \ldots, 0, X_{j_2}, 0, \ldots, 0).$$

(4.8) alapján látható, hogy $g_0(X_{j_1}, X_{j_2}) = U_0(X_{j_1}, X_{j_2})V_0(X_{j_1}, X_{j_2})$ a $g_0$ egy nem triviális faktorizációja.

Ugyanakkor, mivel

$$\begin{aligned} g_0(X_{j_1}, X_{j_2}) &= b_0(X_{j_1}, X_{j_2}) + a_0(X_{j_1}, X_{j_2}) + \gamma = \\ X_{j_2}^n &\left[ b_0\left(\frac{X_{j_1}}{X_{j_2}}, 1\right) + \frac{1}{X_{j_2}^{n-k}} a_0\left(\frac{X_{j_1}}{X_{j_2}}, 1\right) + \gamma \frac{1}{X_{j_2}^n} \right] \end{aligned} \tag{4.9}$$

a $g_0$ fenti nem triviális faktorizációja a

$$Y^n + A(X)Y^{n-k} + B(X)$$

polinom nem triviális faktorizációjára vezet, ahol $X = \frac{X_{j_1}}{X_{j_2}}, Y = \frac{1}{X_{j_2}}$, $A(X) = \frac{1}{\gamma}a_0(X,1)$ és $B(X) = \frac{1}{\gamma}b_0(X,1)$. Mivel ez a 4.6 lemma alapján lehetetlen, ellentmondásra jutunk és ez bizonyítja a 4.7 Lemmát. □

**4.1. Tétel.** *Legyen az $f(\underline{X}) \in \mathbb{F}_q[X_1, \ldots, X_m]$ polinom olyan, hogy megfelel a 4.7 Definíciónak. Jelölje $N(f, \gamma, q)$ az $f(x_1, \ldots, x_m) = \gamma$ egyenlet megoldásainak számát $x_1, \ldots, x_m \in \mathbb{F}_q$ esetén. Ekkor*

$$|N(f,\gamma,q) - q^{m-1}| \leq (n-1)(n-2)q^{m-3/2} + 5n^{13/3}q^{m-2}. \tag{4.10}$$

*Továbbá, ha $q > 15n^{13/3}$, akkor*

$$|N(f,\gamma,q) - q^{m-1}| \leq (n-1)(n-2)q^{m-3/2} + (5n^2+n+1)q^{m-2}. \tag{4.11}$$

*Bizonyítás.* A 4.7 Lemma alapján következik, hogy a $f(\underline{X}) - \gamma$ polinom abszolút irreducibilis $\mathbb{F}_q$ felett.

Ezért tehát az eredmény következik a 4.4 és 4.5 Lemmákból. □

### 4.4.1. Lavinahatás

Ebben a részben a fenti hash függvény lavinahatására vonatkozó elméleti megállapítások kerülnek ismertetésre. A szóban forgó tétel eddig még publikálatlan saját eredmény. A blokk kódolók elméletében használatos lavinahatás fogalmát a hash függvényekre is értelmezhetjük:

**4.9. Definíció.** *Egy $f$ függvény rendelkezik a szigorú lavinahatás tulajdonságával, ha bármely bemeneti bit megváltoztatása esetén minden kimeneti bit $\frac{1}{2}$ valószínűséggel változik meg.*

Az UDHash kapcsán is feltehetjük a kérdést, hogy a lavinahatás tekintetében milyen tulajdonságokkal rendelkezik. A vonatkozó tétel bizonyításához szükségünk lesz a következő eredményekre:

**4.8. Lemma.** *Legyen $q = p^k$ nem negatív egész, és $f \in \mathbb{F}_q[X]$ egy trinom, ami a következő alakú: $f(X) = X^{p^n} - aX - b$, ahol $a \in \mathbb{F}_q^*$. Legyen $d = gcd(n, k)$ és $m = k/d$. Legyen $Tr_d$ a nyom függvény az $\mathbb{F}_q$ testről a $\mathbb{F}_{q^d}$ testre. Minden $0 \leq i \leq m-1$ értékre definiáljuk a $t_i = \sum_{j=i}^{m-2} p^n(j+1)$ összegeket. Legyen $\alpha_0 = a$ és $\beta_0 = b$. Ha $m > 1$, akkor minden $1 \leq r \leq m-1$ értékre, legyen $\alpha_r = a^{1+p^n+\ldots+p^{nr}}$ és*

$$\beta_r = \sum_{i=0}^{r} a^{s_i} b^{p^{ni}},$$

*ahol $s_i = \sum_{j=i}^{r-1} p^{n(j+1)}$ minden $0 \leq i \leq r-1$ és $s_r = 0$ értékre.*

*Az $f$ trinomnak nincs gyöke az $\mathbb{F}_q$ testben pontosan akkor, ha $\alpha_{m-1} = 1$ és $\beta_{m-1} \neq 0$. Ha $\alpha_{m-1} \neq 1$ akkor az $f$ trinomnak egyetlen gyöke van az $x \in \mathbb{F}_q$ testben, nevezetesen, $x = \beta_{m-1}/(1 - \alpha_{m-1})$. Egyébként az $f$ trinomnak $p^d$ gyöke van az $\mathbb{F}_q$ testben, amelyek $x + \delta\tau$ formában állnak elő, ahol $\delta \in \mathbb{F}_{p^d}$, $\tau$ az $\mathbb{F}_q$ test egy fix eleme, amelyre teljesül, hogy $\tau^{p^n-1} = a$ és bármely $c \in \mathbb{F}_q^*$ elemre, amelyre teljesül, hogy $Tr_d(c) \in \mathbb{F}_{p^d}$,*

$$x = \frac{1}{Tr_d(c)} \sum_{i=0}^{m-1} \left( \sum_{j=0}^{i} c^{p^{nj}} \right) a^{ti} b^{p^{ni}}$$

*érvényes.*

*Bizonyítás.* Ez a 3. Tétel a [19] cikkben. □

**4.9. Lemma.** *Definiáljuk az $f \in \mathbb{F}_{2^k}[x_1, \ldots, x_m]$ polinomot, mint $f(x_1, \ldots, x_m) = \sum_{i=1}^{m} \alpha_i x_i^n + \sum_{i=1}^{m} \beta_i x_i$, ahol $n = 2^l + 1$ olyan, hogy $(l, k) = 1$. Legyen $Tr$ az abszolút nyom függvény $\mathbb{F}_q$ felett. Az $f(x_1, \ldots, x_m) - f(x_1, \ldots, x_j + \delta, \ldots, x_m) = \gamma$ egyenlőség pontosan akkor érvényes, ha $Tr((\beta_j\delta+\gamma)\alpha_j^{-1}\delta^{-n}+1) = 0$, és kizárólag két különböző $x_j$ értékre.*

*Bizonyítás.*

$$f(x_1, \ldots, x_m) = \sum_{i=1}^{m} \alpha_i x_i^n + \sum_{i=1}^{m} \beta_i x_i,$$

$$f(x_1,\ldots,x_j+\delta,\ldots,x_m) = \sum_{\substack{i=1\\ i\neq j}}^{m} \alpha_i x_i^n + (x_j+\delta)^n\alpha_j + \sum_{\substack{i=1\\ i\neq j}}^{m} \beta_i x_i + (x_j+\delta)\beta_j,$$

$$f(x_1,\ldots,x_m) - f(x_1,\ldots,x_j+\delta,\ldots,x_m) = \alpha_j(x_j^n + (x_j+\delta)^n) + \beta_j\delta.$$

Következésképp, $f(x_1,\ldots,x_m) - f(x_1,\ldots,x_j+\delta,\ldots,x_m) = \gamma$ pontosan akkor teljesül, ha $x_j$ értéke a következő polinom zérushelye:

$$p(x) = x^n + (x+\delta)^n + \gamma',$$

ahol $\gamma' = (\beta_j\delta+\gamma)\alpha_j^{-1}$. Mivel $p(x) = \delta^n(y^n+(y+1)^n+\gamma'')$, ahol $y = x\alpha_j^{-1}$ és $\gamma'' = \gamma'\delta^{-n}$ a

$$p'(y) = y^n + (y+1)^n + \gamma''$$

zérushelyeit kell meghatároznunk. Mivel $n = 2^l + 1$

$$p'(y) = y^n + (y+1)(y^{n-1}+1) + \gamma'' = y^{2^l} + y + (\gamma''+1).$$

A 4.8 Lemma szerint, ha $(l,k) = 1$, akkor a $p'(y)$ polinomnak vagy 2, vagy pedig 0 gyöke van a $\gamma''$ értékétől függően. Mivel $(l,k) = 1$, $a = 1$ és $b = \gamma''+1$

$$\beta_{k-1} = \sum_{i=0}^{k-1} (\gamma''+1)^{2^{li}}.$$

Az $1,\ldots,k-1$ egészek teljes maradékosztályt alkotnak modulo $k$. $(l,k) = 1$ következésképp $l,\ldots,(k-1)l$ szintén teljes maradékosztály modulo $k$. Mivel $\delta^{2^k} = \delta$ teljesül minden $\delta \in \mathbb{F}_{2^k}$ értékre,

$$\beta_{k-1} = \sum_{i=0}^{k-1} (\gamma''+1)^{2^i} = Tr((\beta_j\delta+\gamma)\alpha_j^{-1}\delta^{-n}+1)$$

következik. □

**4.2. Megjegyzés.** *A szóban forgó $f$ polinom nyilvánvalóan megfelel a 4.7 Definíciónak.*

**4.3. Megjegyzés.** *Ahhoz, hogy az UDHash rendelkezzen a lavinahatás tulajdonságával a szigorúan vett értelemben, $p''(y) = y^n+(y+1)^n$ permutációs polinom kellene, hogy legyen $\mathbb{F}_q$ felett. Mivel $p''(y)$ nyilvánvalóan nem permutációs polinom $\mathbb{F}_2$ felett, ezért nem lehet az $\mathbb{F}_q$ felett sem.*

Habár az UDHash nem rendelkezik a lavinahatás tulajdonsággal a szigorúan vett értelemben, egy valamivel gyengébb állítás érvényes:

**4.2. Tétel.** *Definiáljuk az $f \in \mathbb{F}_{2^k}[x_1, \ldots, x_m]$ polinomot, mint $f(x_1, \ldots, x_m) = \sum_{i=1}^{m} \alpha_i x_i^n + \sum_{i=1}^{m} \beta_i x_i$, ahol $n = 2^l + 1$ olyan, hogy $(l, k) = 1$. Ekkor*

$$(1-q\varepsilon)^{m-1}(\frac{1}{q}-\varepsilon) \leq P(f(x_1, \ldots, x_m)-f(x_1+\delta_1, \ldots, x_m+\delta_m) = \gamma) \leq (1+q\varepsilon)^{m-1}(\frac{1}{q}+\varepsilon)$$

*, ahol $0 \leq \varepsilon \leq nq^{-\frac{3}{2}}$.*

*Bizonyítás.* Legyen $D_{\gamma_i}$ az az esemény, hogy $f(x_1, \ldots, x_m) - f(x_1, \ldots, x_i + \delta_i, \ldots, x_m) = \gamma_i$. A 4.9 Lemma alapján:

$$P(D_{\gamma_i}|\delta_i \in A_{\gamma_i}) = 0, \quad P(D_{\gamma_i}|\delta_i \in B_{\gamma_i}) = \tfrac{1}{2^{k-1}},$$

ahol $A_{\gamma_i} = \{\delta | Tr((\beta_i\delta+\gamma)\alpha_i^{-1}\delta^{-n}+1) = 1\}$, $B_{\gamma_i} = \{\delta | Tr((\beta_i\delta+\gamma)\alpha_i^{-1}\delta^{-n}+1) = 0\}$. Mivel $g(x) = (\beta_j x + \gamma)\alpha_j^{-1}x^{-n} + 1 = \alpha_j^{-1}\beta_j x^{1-n} + \gamma\alpha_i^{-1}x^{-n} + 1$. Legyen $h(x) = \alpha_j^{-1}\beta_j x^{n+1}+\gamma\alpha_i^{-1}x^n+1$. Mivel $\chi(g(x)) = \chi(h(x^{-1}))$ minden $x \neq 0,\ x \in \mathbb{F}_q$ esetén és $\chi(g(0)) = \chi(h(0))$, a Weil tétel alapján (3.5 Lemma)

$$||A_{\gamma_i}| - |B_{\gamma_i}|| = \left|\sum_{x\in\mathbb{F}_q} \chi(g(x))\right| = \left|\sum_{x\in\mathbb{F}_q} \chi(h(x))\right| \leq nq^{1/2}$$

következik. $P(\delta_i \in A_{\gamma_i}) + P(\delta_i \in B_{\gamma_i}) = 1$, következésképp

$$P(A_{\gamma_i}) = \tfrac{1}{2} \pm \epsilon_{\gamma_i} \quad P(B_{\gamma_i}) = \tfrac{1}{2} \mp \epsilon_{\gamma_i}\ ,$$

ahol $\epsilon_{\gamma_i} \leq \frac{n}{2q^{1/2}}$. A teljes valószínűség tétele szerint

$$P(D_{\gamma_i}) = P(D_{\gamma_i}|A_{\gamma_i})P(A_{\gamma_i}) + P(D_{\gamma_i}|B_{\gamma_i})P(B_{\gamma_i}) = \frac{1}{q} \pm \varepsilon_{\gamma_i},$$

ahol $\varepsilon_{\gamma_i} \leq nq^{-3/2}$. Vegyük észre, hogy a $f(x_1, \ldots, x_m)$ polinom szerkezete miatt, a $D_{\gamma_i}$ események függetlenek, továbbá $f(x_1, \ldots, x_m) - f(x_1 + \alpha_1, \ldots, x_m + \alpha_m) = \gamma$ pontosan akkor teljesül, ha a $D_{\gamma_i}$ $(i = 1, \ldots, m)$ események teljesülnek és $\gamma_1 + \ldots + \gamma_m = \gamma$. Ezért

$$P(f(x_1, \ldots, x_m) - f(x_1 + \alpha_1, \ldots, x_m + \alpha_m) = \gamma) = \sum_{\substack{i_1, \ldots, i_m \\ \gamma_{i_1} + \ldots + \gamma_{i_m} = \gamma}} \prod_{j=1}^{m} P(D_{\gamma_{i_j}}).$$

A 3.11 Lemma szerint $h(x_1, \ldots, x_m) = x_1 + \ldots + x_m$ permutációs polinom, és mint ilyennek $q^{m-1}$ megoldása van. Következésképp

$$\sum_{\substack{i_1, \ldots, i_m \\ \gamma_{i_1} + \ldots + \gamma_{i_m} = \gamma}} \prod_{j=1}^{m} P(D_{\gamma_{i_j}}) \leq q^{m-1}(\frac{1}{q} + \varepsilon)^m = (1 + q\varepsilon)^{m-1}(\frac{1}{q} + \varepsilon)$$

és

$$\sum_{\substack{i_1, \ldots, i_m \\ \gamma_{i_1} + \ldots + \gamma_{i_m} = \gamma}} \prod_{j=1}^{m} P(D_{\gamma_{i_j}}) \geq q^{m-1}(\frac{1}{q} - \varepsilon)^m = (1 - q\varepsilon)^{m-1}(\frac{1}{q} - \varepsilon)$$

érvényesek, ahol $\varepsilon = \max_{\gamma_i} \varepsilon_{\gamma_i}$ és a maximumot az összes lehetséges $\gamma_i$ vektor felett vesszük, amely $h(x_1, \ldots, x_m)$ megoldásaként megjelenik. □

# 5. fejezet

# Implementáció

Ebben a fejezetben az előzőekben ismertetett UDHash függvény implementációjáról és a DESignIn beléptetőrendszerről lesz szó. Az UDHash egy iteratív hash függvény és a korábbiakban csak a tömörítőfüggvény került ismertetésre. Az első szakasz az UDHash paraméterválasztásáról és az alkalmazott iterációs eljárásról szól. A második szakasz témája a konkrét implementációs kérdések, a véges test kiválasztása, továbbá a futási időre és a lavinahatásra vonatkozó tesztek. A harmadik szakaszban, mint lehetséges alkalmazási terület, a DESignIn azonosító rendszer kerül ismertetésre. A DESignIn tervezési megfontolásai közös eredményeink Huszti Andreával és Pethő Attilával és a [26] cikkben kerültek publikálásra. A második szakaszban leírt gyakorlati vizsgálatok eredményei eddig nem lettek publikálva és a B illetve a D Appendixben találhatóak.

## 5.1. UDHash a gyakorlatban

Az előző fejezetben ismertetett UDHash függvény a gyakorlatban is implementálásra került. A következőkben ezen implementációnak a paraméterválasztása és az ezzel kapcsolatos megfontolások kerülnek ismertetésre.

A legtöbb kriptográfiai primitív esetén a szempont az, hogy gyorsan számíthatóak legyenek. A hash függvények esetén azonban az is szem-

pont lehet, hogy a leggyorsabb implementáció is viszonylag lassú legyen. A számítási erőforrások manapság olcsók és skálázhatóak, ezért a gyors hash függvények esetén a feltöréssel próbálkozóknak is könnyebb dolga van. Thomas Roth az Amazon Elastic Compute Cloud szolgáltatásának segítségével tört fel erős kriptográfiai hash függvénnyel védett jelszavakat olcsó wifi készülékeken [1] 20 perc alatt, percenként 28 centes költséggel. Az UDHash ebből a szempontból is előnyös, hiszen a leggyorsabb implementációja is lassabb, mint a napjainkban gyakorlatban használt hash függvényeké.

### 5.1.1. Paraméterválasztás

A véges test megválasztásánál tipikusan két utat szokás mérlegelni: az egyik esetben $q$ egy prím, míg a másikban a 2 valamilyen hatványa. A kimerítő kulcstámadás elkerüléséhez a $q$ legalább 128 bit nagyságú kell hogy legyen.

Az UDHash függvény a paraméterek széles választékára ad lehetőséget. A gyakorlatban olyan polinomok osztályára van szükségünk, amelyre teljesülnek a 4.7 Definíció feltételei, ugyanis csak ezen feltétel mellett érvényesek a 4.1 Tétel eredményei.

Ennek megfelelően az implementációban alkalmazott polinomok a következő eredmény alapján kerültek kiválasztásra [7]:

**5.1. Tétel.** *Legyen* $f(\underline{X}) = b(\underline{X}) + a(\underline{X})$ *olyan, hogy* $b(\underline{X}) = \beta_1 X_1^r + \cdots + \beta_m X_m^r$, $a(\underline{X}) = \alpha_1 X_1^s + \cdots + \alpha_m X_m^s$ *és* $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m \neq 0$. *Ha* $0 < s < r < q$ *és* $r$ *páratlan, ha* $q = 2^f$, *akkor* $f(\underline{X})$ *teljesíti a 4.1 Definíció feltételeit.*

*Bizonyítás.* Az $f(\underline{X})$ megválasztásával a 4.1 Definíció feltételei automatikusan teljesülnek, azon feltétel kivételével, hogy a $b_0(X_i, X_j) = \beta_i X_i^r + \beta_j X_j^r$ polinomnak ne legyen többszörös gyöke.

A $b_0(X_i, X_j)$ polinomnak pontosan akkor van többszörös gyöke $\bar{\mathbb{F}}_q$ felett, ha a $c(\underline{X}) = \underline{X}^r + \gamma$ polinomnak $\underline{X} = X_i/X_j$ és $\gamma = \beta_j/\beta_i$ esetén többszörös gyöke van $\bar{\mathbb{F}}_q$ felett. $c(\underline{X})$ többszörös gyökei a $\gcd(c(\underline{X}), c'(\underline{X}))$ polinomnak is gyökei. $c'(\underline{X}) = r\underline{X}^{r-1}$, csak akkor nem nulla, ha $r$ és $\mathbb{F}_q$ karakterisztikája relatív prímek. Ez érvényes minden $r$ esetén, ha $q$ prím, és minden páratlan $r$ esetén, ha $q = 2^f$. Továbbá, ha $c'(\underline{X}) \neq 0$, akkor az egyetlen gyöke 0,

ami akkor és csakis akkor gyöke a $c(\underline{X})$ polinomnak, ha $\gamma = 0$, de ez a $\beta_i$ értékek megválasztása miatt nem lehetséges. □

### 5.1.2. Iterációs séma

A CodeFish elleni támadások egyik alapja, az alkalmazott iterációs séma gyengesége volt. Az UDHash esetében ennek kiküszöbölésére a Merkle-Damgård konstrukció került alkalmazásra (4.6. Algoritmus a [68] könyvben):

**5.1. Algoritmus** (Merkle-Damgård séma)**.** *A bemeneten kapott $x$ bitsztringhez rendel hozzá lenyomatot:*

$$\{0,1\}^{m+t} \to \{0,1\}^m, \textit{ ahol } t \geq 2$$

1. $n \leftarrow |x|$
2. $k \leftarrow \lceil n/(t-1) \rceil$
3. $d \leftarrow k(t-1) - n$
4. ***for*** $i \leftarrow 1$ ***to*** $k-1$
   ***do*** $y_i \leftarrow x_i$
5. $y_k \leftarrow x_k || 0^d$
6. $y_{k+1} \leftarrow d$ *bináris reprezentációja*
7. $z_1 \leftarrow 0^{m+1} || y_1$
8. $g_1 \leftarrow$ ***compress***$(z_1)$
9. ***for*** $i \leftarrow 1$ ***to*** $k$
   ***do*** $\begin{cases} z_{i+1} \leftarrow g_i || 1 || y_{i+1} \\ g_{i+1} \leftarrow \textbf{\textit{compress}}(z_{i+1}) \end{cases}$
10. $h(x) \leftarrow g_{k+1}$

A fenti algoritmusban az $|x|$ az $x$ bitsztring hosszát jelöli, az $x||y$ pedig az $x$ és az $y$ bitsztringek konkatenációját. A $0^k$ egy $k$ hosszú, csupa nullából álló bitsztring és a **compress** pedig a tömörítőfüggvényt jelöli. A fenti sémát használó iteratív hash függvényekre igaz a következő állítás:

**5.1. Lemma.** *Tegyük fel, hogy* ***compress****:* $\{0,1\}^{m+t} \to \{q,1\}^m$ *ütközésellenálló tömörítő függvény, ahol* $t \geq 2$*. Ekkor a 5.1 Algoritmus alapján számított*

$$h : \bigcup_{i=m+t+1}^{\infty} \{0,1\}^i \to \{0,1\}^m,$$

*függvény ütközésellenálló.*

*Bizonyítás.* Ez a 4.6. Tétel a [68] könyvben. □

Ebből ugyan nem következik, hogy az ily módon nyert hash függvény megőrzi a tömörítőfüggvény előző fejezetben ismertetett jó tulajdonságait, de ebbe az irányba mutat. Pontos elméleti eredmények híján a megvalósításkor csupán ez az eredmény alapozta meg azt a feltételezést, miszerint az iterációs séma nem visz a konstrukcióba a [6] cikkben ismertetetthez hasonló gyengeséget.

## 5.2. Az UDHash implementációja

Az UDHash megvalósításakor nagy jelentőséggel bírnak a véges testek felett végzett műveletek algoritmusai. Ezen műveletek számítási ideje nagyban függ a választott véges testtől. A test megválasztása ezért jelentősen befolyásolja az egyes algoritmusok teljesítményét.

A test mérete nyilvánvalóan hatással van a tényleges számítási időre. A választott test karakterisztikája azonban meghatározza a felhasználható algoritmusokat és ezáltal az aszimptotikus számítási időt fogja befolyásolni.

A választott test karakterisztikáján múlik, hogy egyszerű moduláris aritmetikát használhatunk, vagy a páros karakterisztikájú testekre vonatkozó algoritmusok valamelyikét kell használnunk. A páros karakterisztika

elsősorban hardver implementáció esetén előnyös, míg a prímtestek alkalmazásával az általános célú processzorokon lehet nagyobb teljesítményt elérni.

### 5.2.1. Prímtestek aritmetikája

A prímtestek feletti műveletvégzés lényegében moduláris aritmetikát jelent. Az általános célú processzorok utasításkészlete rendszerint tartalmazza az egészekkel való műveletvégzéshez szükséges utasításokat. A konstrukcióink biztonságához szükséges, hogy nagy testek felett dolgozzunk, ez azzal jár, hogy a processzoraink szóhosszánál lényegesen hosszabb adatokkal kell dolgoznunk.

A nagy számokkal gyorsan kell műveleteket végeznünk, ehhez az általános szorzóalgoritmusok, mint a Karatsuba szorzás (8.1. Algoritmus a [74] könyvben), FFT szorzás (8.16. Algoritmus a [74] könyvben), a Toom-Cook algoritmus (4.3.3. szakasz a[45] könyvben) vagy a gyors maradékos osztás (9.5. Algoritmus a [74] könyvben) jól használhatóak.

A nagy számokkal való műveletvégzés egy nagyon általános igény, így a tetszőleges hosszúságú aritmetika megvalósítására sok különböző programozási nyelvhez léteznek programkönyvtárak. Ezek többnyire alaposan kitesztelt, jól optimalizált implementációk, amik rendszerint alacsony szintű, platformonként specifikus, assembly kódbetéteket is bevetnek a nagyobb sebesség elérésének érdekében.

A prímtestek feletti aritmetika megvalósítására a GNU Multiprecision programkönyvtárat (http://gmplib.org/) használtuk. A GMP a használt algoritmusok megválasztásához egy lépcsős megközelítést alkalmaz. Az operandusok méretétől függően azt az algoritmust használja, amely abban a tartományban a leggyorsabb. A Karatsuba, az FFT és a Toom szorzóalgoritmus több változata közül választja ki a leginkább megfelelőt. Az egyes algoritmusok sebessége és így a határok is függenek a platformtól és a GMP alapértelmezett működése ezt a tényezőt is kezeli.

### 5.2.2. Páros karakterisztikájú aritmetika

Az általános célú processzorok utasításai nem támogatják olyan szinten a páros karakterisztikájú véges testek feletti műveletvégzést, mint ahogyan a prímrendű testek esetében. A páros karakterisztika előnyei elsősorban hardverimplementáció esetén mutatkoznak meg.

Páros karakterisztika esetén a teljesítményt befolyásoló újabb kérdés a testek elemeinek a reprezentációja. Polinomreprezentáció esetén a Montgomery szorzás [17] gyors műveletvégzést tesz lehetővé.

Polinomreprezentáción alapszik az MPFQ (http://mpfq.gforge.inria.fr/) programkönyvtár is. Ez kettő karakterisztikájú prímtestek műveleteit is implementálja. Egyedi megközelítést alkalmaz abban a tekintetben, hogy nem algoritmusokat definiál és alkalmaz, hanem az adott test paramétereinek az ismeretében speciálisan arra a testre szabott programkódot generál, ezáltal a specifikus gyorsítási lehetőségeket is ki tudja aknázni az egyes testek esetén.

A normál bázis reprezentáció vonzó tulajdonsága, hogy a négyzetre emelés lényegében ciklikus eltolással egyenértékű és ezért nagyon gyors. A normál bázis szorzás kiterjedt tanulmányozás tárgya volt az utóbbi időben és több algoritmus is született a legkülönfélébb architektúrákra [2][3][40][51][70]. Ezek hardver implementáció esetén gyors műveletvégzést tesznek lehetővé. Általános célú processzorok esetén sajnos a szorzás annyival lassabb, hogy a legtöbb esetben nem éri meg a normál bázis reprezentáció mellett dönteni.

A normál bázis szorzóalgoritmusok sok bitszintű műveletet használnak és ezért nem tudják teljes mértékben kihasználni a processzor adatútvonalait. A [61] cikkben a szerzők egy olyan algoritmust javasoltak, amely kiküszöböli ezt a problémát és gyorsabb normál bázis szorzást tesz lehetővé. Az UDHash implementációjában a [61] cikkben szereplő 2. Algoritmust használtuk fel:

**5.2. Algoritmus.** *(Szorzás páros típusú GNB esetén)*
*Input:* $A, B \in \mathbb{F}_{2^m}, \Delta F(x) \in [0, m-1], 1 \leq n \leq p-1$
*Output:* $C = AB$

1. $S_A \leftarrow A, S_B \leftarrow B, C \leftarrow 0$

2. ***for*** $n = 1$ ***to*** $p - 2$

$$\textbf{\textit{do}} \begin{cases} S_A \ll \Delta F(n) \\ R \leftarrow S_A \odot S_B \\ C \leftarrow C + R \\ S_B \ll \Delta F(n) \end{cases}$$

Ahol a $\Delta F(x)$ a

$$F(2^i u^j \mod p) = i, \;\; 0 \leq i \leq m-1, \; 0 \leq j < t$$

előszámolt értékeket jelenti, ahol az $u$ egy $t$ rendű elem $\mod p$. Az algoritmus leírásában a $\odot$ a bitenkénti AND műveletet a $\ll$ a ciklikus balra léptetést a $+$ pedig a bitenkénti XOR műveletet jelenti.

Az algoritmus csak páros típusú Gauss normál bázisok esetén alkalmazható (Gaussian Normal Base - GNB). A Gauss normál bázisok speciális, alacsony bonyolultságú normál bázisok. Az $\mathbb{F}_{2^k}$ testre akkor létezik Gauss normál bázis, ha a $k$ nem osztható nyolccal. Pontosan akkor létezik a testhez $t$ típusú Gauss normál bázis, ha $p = tk + 1$ prím és $\gcd(\frac{tk}{l}, k) = 1$), ahol $l$ a 2 multiplikatív rendje modulo $p$.

### 5.2.3. Futási idő

Az implementáció során a pártlan karakterisztikájú testek esetén a GNU MP programkönyvtár szolgál az aritmetika megvalósítására, páros karakterisztika esetén pedig az előző alszakaszban ismertetett 5.2 Algoritmus alapján készült saját függvény került felhasználásra. A hash függvény mindkét esetben a 5.1 Tételnek megfelelő polinomválasztással lett paraméterezve, mégpedig úgy, hogy a polinom mindkét komponensének három tagja van. A mérési eredmények egy véletlenszerűen generált 100 Megabájtos fájl hashelésével születtek. A többi hash függvényre vonatkozó mérések a Crypto++ programkönyvtár implementációjával készültek. A tesztek pontos hardver- és szoftverkörnyezetének leírása az A Appendixben található, a mért UDHash függvények pontos paraméterezése a C Appendixben, a mérési eredmények pedig a D Appendixben kerülnek közlésre.

A mérések alapján kiderül, hogy az UDHash leggyorsabb implementációi is nagyságrendekkel lassabbak, mint a többi hash függvény. Ez megoldást jelenthet arra a problémára, hogy napjainkban bárki számára olcsón hozzáférhető nagy számítási kapacitás, és hogy ez a gyors hash függvényeket sebezhetővé teszi (lásd [1]).

### 5.2.4. Lavinahatás

A 4.2 Tétel feltételei sajnos túl szigorúak ahhoz, hogy a gyakorlati megvalósítás esetén ezeknek megfelelő paramétereket alkalmazzunk. Ezért a konstrukció lavinahatás tulajdonságára vonatkozóan gyakorlati vizsgálatok is folytak.

A hash függvény egyes paraméterezései mellett 1500-1500 véletlenszerűen választott mintára lett megvizsgálva, hogy az egyes bemeneti bitek megváltoztatása, milyen hatással van az egyes kimeneti bitekre, azaz, hogy lényegében a lavina hatás tulajdonság tekintetében a gyakorlatban hogyan viselkedik a függvény.

A vonatkozó teszteredmények az B Függelékben kerülnek közlésre. Itt az egyes ábrák alatt a hash függvény paraméterezése a lineáris és a nemlineáris tagok együtthatóival, illetve a nemlineáris tag kitevőjével van megadva. A grafikon $x$ tengelyén található számok 0-761 -ig az egyes bemeneti biteket jelölik, az $y$ tengely számozása 0-243-ig a kimeneti bitekhez tartozik. Az $z$ tengely pedig 0-1499-ig a bemeneti minták számát jelöli. Egy (x,y,z) pont az ábrán azt jelenti, hogy a bemenet $x$. bitjét megváltoztatva a kimenet $y$. bitje $z$ darab minta esetében változott meg.

Látható, hogy a tesztek esetében a pontok a $z = 750$ sík környezetében találhatóak, azaz az empirikus valószínűség minden esetben a lavinahatás tulajdonságtól elvárt $\frac{1}{2}$ közelében van. Azaz, habár az UDHash a fent választott paraméterezés mellett, szigorúan vett értelemben bizonyíthatóan nem rendelkezik a lavinahatás tulajdonságával (4.3 Megjegyzés), a gyakorlatban egy, a lavinahatáshoz nagyon közeli viselkedést mutat.

## 5.3. UDSignIn

A hash függvények és az álvéletlen-szám generátorok számos kriptográfiai protokollban fontos szerepet játszanak. A jelen dolgozatban ismertetett kriptográfiai primitívek lehetséges alkalmazási területének egy példája az UDSignIn azonosító rendszer.

Az UDSignIn egy univerzális webes alapú azonosítórendszernek lett tervezve. A feladata az egyszerű jelszavas bejelentkezés kiváltása, kényelmes megoldást kínál arra a problémára, hogy a biztonságos jelszavakat rendszerint nehéz megjegyezni, és hogy manapság egy átlagos felhasználó több tucat webes accounttal is rendelkezhet.

A jelszavas azonosítás esetén a biztonság érdekében a felhasználóknak minden egyes accounthoz más, erős, azaz nehezen megjegyezhető jelszót kellene észben tartania. Ez rendszerint ahhoz vezet, hogy gyenge jelszavakat használnak, vagy, hogy több helyen is ugyanazt a jelszót használják, esetleg felírják valahová. A jelszavas azonosítás felváltása tanúsítványok segítségével való azonosításra nem csak a rendszer biztonságát növeli, de a felhasználóknak is kényelmesebb megoldást jelent.

Az UDSignIn ezt a problémát egy hardvertoken segítségével hidalja át. Az azonosításhoz aszimmetrikus kriptográfiát használ, a szerver azonosításában már meglévő, széles körben elterjedt és bevált technikára, konkrétan az SSL egyik változatára támaszkodik. Az UDSignIn technikai leírása, és a vele kapcsolatos elméleti, gyakorlati és tervezési megfontolások a [26] cikkben kerültek publikálásra.

### 5.3.1. SSL hibrid

A világhálón a titkosított adatközlésre széles körben elterjedt protokoll a https, amely az SSL (Secure Socket Layer) titkosító réteg valamely változatát használja. A jelszavas bejelentkező felületek is rendszerint kihasználják a https előnyeit. Ugyan a https lehetővé teszi a kölcsönös azonosítást, egy az UDSignIn-hez hasonló protokollal, amely része az SSL protokollkészletének, ennek a használata azonban az átlag felhasználó számára meglehetősen körülményes.

Az SSL-t a Netscape tervezte és implementálta először. A jelenlegi verziót (3.0) 1996-ban adták ki [66]. Az SSL protokoll lehetővé teszi a szerver, és opcionálisan a kliens azonosítását és a titkosított kommunikációt.

Az SSL az internetes protokollok két rétege közé épül be, külön rétegnek is tekinthető, ezért nem csak a http, de bármely alkalmazásszintű protokoll titkosítására használható.

A kliens és a szerver azonosítása aszimmetrikus kriptográfia és publikus kulcsú infrastruktúra (PKI) segítségével történik. A kliens azonosításáról https esetében a böngésző gondoskodik. A módszer problémája a nehézkes konfiguráción kívül, hogy a tanúsítvány tárolása is erősen platformfüggő (az Internet Explorer például a registryben tárolja). A kettő a mobilitást a konfigurált gépekre korlátozza és nem teszi lehetővé az egyszerű és biztonságos hozzáférést bármely számítógépről.

Az UDSignIn áthidalja ezt a problémát és a https azonosítását egy könnyen kezelhető és hordozható, szintén PKI alapú kliensazonosítással váltja ki.

### 5.3.2. Kliens implementáció

A cél egy könnyen használható és mobilis azonosítás megvalósítása és az SSL kliensazonosításának kiváltása. A probléma három különböző szinten oldható meg:

**Szállítási réteg** Ezen a szinten szükséges az SSL protokoll módosítása, oly módon, hogy a fenti feltételeknek megfeleljen. Ennek mind elméleti, mind gyakorlati akadályai is vannak:

- Ha az SSL-t a szállítási réteg tetejének tekintjük, akkor a felhasználóazonosítás logikailag nem illik bele a rétegbe. (Az SSL PKI alapú kliensazonosítása jellegénél fogva elsősorban eszközök azonosítására alkalmas, és elméleti szempontból is arra kellene, hogy szolgáljon.)
- Valódi hordozhatóságot kívánunk elérni. Ha az aktuális protokollt módosítjuk, a használatba vétel előtt el is kell terjeszteni azt a változatot.

**Alkalmazási réteg (Böngésző szint)** Megváltoztathatjuk a böngésző tanúsítványkezelési lehetőségeit (ezt tipikusan böngésző beépülőkkel érhetjük el). A fő probléma ezzel a megközelítéssel, hogy a valódi hordozhatósághoz minden elterjedtebb böngészőre implementálni kell az azonosítást, továbbá, hogy nem feltétlenül van joga egy általános felhasználónak egy tetszőleges gépen új böngésző beépülőket telepíteni. Ugyanakkor ez azt is jelenti, hogy ez a módszer szintén a számítógép előzetes felkészítését igényli.

**Alkalmazási réteg (Legfelső szint)** A harmadik lehetőség, hogy az új funkciókat kliensoldali kód segítségével implementáljuk. A hátránya ennek a módszernek, hogy itt nem használhatjuk közvetlenül az SSL protokoll kliensazonosítási mechanizmusát, mert annak a rétegnek a szolgáltatásait már elrejti a böngésző. Ebben az esetben egy teljesen új módszert kell az azonosításra használnunk.

A harmadik esetben nincs nagyobb gyakorlati akadály, ezért az UDSignIn esetében is ezt választottuk. Az implementációnak képesnek kell lennie elérni bizonyos rendszererőforrásokat, hogy hozzáférjen a tanúsítványhoz, és a lehető legtöbb platformon elérhetőnek kell lennie. Több különböző alternatíva áll rendelkezésre a kliens oldali kódot illetően:

**Java applet** Ha a kliens gépen van Java Runtime Environment (JRE) telepítve, akkor annak segítségével Java alkalmazásokat futtathatunk a böngészőben. Biztonsági okokból a böngészők korlátozzák a program hozzáférését az erőforrásokhoz (Java Sandbox). Ahhoz, hogy a feladat ellátásához szükséges erőforrásokhoz hozzáférjünk, alá kell írnunk a Java kódot, amivel annak eredetét bizonyítjuk, és felelősséget vállalunk érte. Ha a felhasználó megbízik a kód gazdájában akkor futtatja, és az hozzáférhet a kívánt erőforrásokhoz, egyébként pedig egyáltalán nem fut le.

**ActiveX** Az ActiveX HTML oldalakba ágyazott bináris kód (ellentétben a Java Applettel és a JavaScripttel, ahol az interpreteres megoldásnak, illetve virtuális gépnek köszönhetően a kód platformfüggetlen). Az

applethez hasonlóan ez is digitális aláírást használ a kód megbízhatóságának igazolására.

**JavaScript** A JavaScript egy szkriptnyelv, amit a böngésző interpretere futtat, ezért erősen függ a használt böngészőtől. Akárcsak a Java Applet ez is lehetővé teszi mind a homokozó modell (Sandbox), mind a digitális aláírással hitelesített kód használatát. Viszont nem minden böngészőben van mind a két modell implementálva: Például a 4-es előtti Netscape verziók és az Internet Explorer egyáltalán nem teszi lehetővé a hitelesítés használatát.

Megfontolandó lehet a három megoldás valamilyen kombinációját alkalmazni platformtól függően, de messze a legerőforráskímélőbb megvalósítás a Java Applet használata, és az UDSignIn esetében is ezt a lehetőséget választottuk.

### 5.3.3. Hardver kulcs

Az UDSignIn használatához csupán egy hardverkulcs szükséges, amit aztán tetszőleges gépen lehet használni egy megadott oldalra való bejelentkezéshez. A hardverkulcs a kliensazonosításhoz szükséges titkos kulcsot tárolja. A hardverkulcs elméletileg lehet smart kártya, USB token, sima USB tároló is. A kártyaolvasók nem számítanak standard felszerelésnek, az USB tokenek esetén pedig az UDSignIn implementációja idején (2006 márciusa) nem állt rendelkezésre tisztán java alapú meghajtó, a böngészők pedig a gyakorlatban nem tették lehetővé a Java Native Interfésszel megvalósított programkönyvtárak használatát a java appleteknek, csak ha azok már előzetesen telepítve voltak a kliensgépre. Ezért a hordozhatóság megőrzése érdekében a kulcs tárolására hétköznapi USB tárolókat használtunk.

### 5.3.4. Protokoll

Az UDSignIn az ISO/IEC 9798-3 protokoll egy módosított változatát használja a kliens azonosítására:

1. A szerver generál egy egyszer használatos véletlen számot. Ezt az $n$ véletlen kérdést egy $t$ időbélyeggel összefűzve eltárolja az adatbázisban és az SSL titkos kulccsal titkosítva elküldi a kliensnek.

$$S \longrightarrow C : Enc_{SK}(n||t)$$

2. A kliens, amint megkapta a szerver üzenetét, az $ID$ azonosítóját és digitális aláírását felhasználva elkészít egy azonosító tokent (a digitális aláíró sémák rendszerint egy kriptográfiai hash függvény segítségével készítenek lenyomatot az aláírandó adatokról).

$$C \longrightarrow S : Enc_{SK}(n||t, ID, Sig_C(n||t))$$

3. A szerver kikeresi az $ID$-hez tartozó publikus kulcsot, és ellenőrzi az aláírást. Ezek után megnézi, hogy az $n||t$ érték szerepel-e az adatbázisban és ellenőrzi, hogy egy meghatározott időlimiten belül került-e oda. Ha valamely ellenőrzés sikertelen, akkor a szerver megszakítja a kommunikációt, egyébként pedig törli az adatbázisból az $n||t$ értéket és biztosítja a felhasználó számára a hozzáférést.

### 5.3.5. Megvalósítás

A megvalósítása szerver oldalon PHP technológiára épül, de lényegében tetszőleges keretrendszerrel kiváltható a funkciója. A lényeges pont ami a rendszer előnyét és egyben hátrányát is jelenti, hogy a kliens oldalon a böngésző által futtatott java applet végzi el a protokollhoz kapcsolódó teendőket.

Ez egyrészt lehetővé teszi, hogy bármely számítógépről előzetes felkészítés nélkül elérhető és használható legyen a rendszer. Ugyanakkor az UDSignIn készítésekor (2006 márciusa) még nem volt elérhető olyan hardvertoken, amely tisztán java interfésszel rendelkezett volna, a böngészőkbe beépülő java futtatási környezet pedig csupán elméletileg tette lehetővé a Java Natív Interfész alkalmazását egy távoli szerverről letöltött programkönyvtár futtatásával.

Ez azt jelenti, hogy az UDSignIn ugyan a tervezési céloknak megfelelően általánosan használható webes bejelentkezési rendszer, de ennek az az ára, hogy a rendszer aszimmetrikus kriptográfiájához szükséges tanúsítványokat alacsonyabb biztonságot jelentő hardverelemeken kell tárolni.

Az UDSignIn bejelentkezési rendszert 2006 márciusa óta használja a Debreceni Egyetem Informatika Karának kari tanácsa a belső dokumentumok biztonságos webes hozzáférhetőségének a biztosítására.

# 6. fejezet

# Összefoglalás

Jelen dolgozat fő témája két kriptográfiai primitív (egy álvéletlenszám generátor és egy hash függvény), illetve azok egy lehetséges alkalmazási területe.

Az első fejezet egy rövid bevezetőt tartalmaz. Elhelyezi a kriptográfiát a modern technika eszközei között, hangsúlyozza annak fontosságát és bevezet a kriptográfia alapfogalmaiba. A bevezetőt követően elhelyezi a tanulmányozni kívánt primitíveket a kriptográfián belül és áttekintést ad a dolgozat egészéről és a benne foglalt eredményekről.

A második fejezet témája az álvéletlenszám generátorok és az álvéletlenség fogalma. Ez a fejezet áttekinti a klasszikus álvéletlenségi teszteket és az álvéletlenség egyes definícióit, majd ezekből kiindulva eljut a Sárközy és Mauduit által bevezetett álvéletlen mértékekig ([55]). A szóban forgó mértékek a Normalitás a Jóleloszlás, illetve a Korrelációs mértékek, amelyek közül (tekintettel arra, hogy a Korrelációs mérték alapján felső korlát képezhető a Normalitás mértékre) a Jóleloszlás és a Korrelációs mértékek egyesítésével kapjuk a Kombinált mértéket. Ezek az álvéletlen mértékek képezik a fő eszközt a dolgozat témájául szolgáló álvéletlen generátor tanulmányozásához.

A harmadik fejezet témája maga a generátor. Cikkükben ([55]) Sárközi és Mauduit maguk is adtak példát jó álvéletlenségi mértékkel rendelkező generátorra, és később, részben más szerzők közreműködésével, több jó

konstrukció is született ([39], [49] [36] [37] [56] [34]). A fejezetben ezek közös tervezési elve, a dupla csavar módszer és az egyik, az új konstrukcióhoz közel álló generátor kerül ismertetésre. Ezek után az új generátor konstrukciója következik, amely mindamellett, hogy jó álvéletlen mértékekkel rendelkezik, szemben az összes korábbi konstrukcióval kettő karakterisztikájú véges testeket vesz alapul.

Itt kerülnek bizonyításra az új konstrukció álvéletlen mértékei is. A kriptográfiában nem csupán álvéletlen sorozatokra, hanem nagy álvéletlen sorozat családokra van szükségünk. Fontos az is, hogy a sorozatok, ne csak külön, hanem együtt is jó tulajdonságokkal rendelkezzenek. Ilyen tulajdonság a család bonyolultság és a lavinahatás. Az új konstrukció ezen tulajdonságai is bizonyítást nyernek.

A fejezet tárgyalja a generátor lineáris bonyolultságát is. Kiderül, hogy nem alkalmas kriptográfiai felhasználásra, ugyanis a lineáris bonyolultsága túl alacsony. Ezzel együtt fény derül egy szoros kapcsolatra a lineárisan visszacsatolt léptetőregiszterekkel. Ez a kapcsolat gyors hardveres implementációt tesz lehetővé, és a jó statisztikai tulajdonságokkal együtt kiválóan alkalmassá teszi a nem kriptográfiai alkalmazásokra. A fejezet utolsó három szakaszának eredményei saját eredmények és a [23] és [25] cikkekben kerültek publikálásra.

A negyedik fejezetben egy új hash függvény konstrukcióról van szó. A fejezet a hash függvényekről szól általánosságban, tartalmazza a hash függvény definícióját és a kriptográfiai hash függvényekkel kapcsolatos elvárások megfogalmazását. Ezután a konstrukció Codefish néven implementált elődjét ismerteti ([10]). Ezen konstrukció (pontosabban annak az implementáláshoz felhasznált módosított speciális esete) kriptanalízise is itt kapott helyet.

Ezek után az új konstrukció és a vele kapcsolatos elméleti eredmények következnek. A konstrukció kiküszöböli elődje hibáit ([6]) és az operandusok méretében is előrelépést jelent. Elméleti megfontolások alapozzák meg a függvény előképellenállóságát és egy jó statisztikai tulajdonsága is bizonyítást nyer. A függvény lavinahatásával kapcsolatos vizsgálatok is itt szerepelnek: habár a függvény implementációban felhasznált változata bizonyíthatóan nem rendelkezik a szigorú lavinahatással, aszimptotikusan

nagyon hasonlóan viselkedik hozzá.

A hash függvény előkép ellenállóságára vonatkozó megállapítások Bérczes Attilával és Pethő Attilával közös eredményeink és a [7] cikkben kerültek publikálásra. A lavinahatásra és a hozzá kapcsolódó aszimptotikus állításra vonatkozó tétel saját eredmény és eddig még nem lett publikálva.

Az ötödik fejezetben az említett hash függvény implementációjáról és a DESignIn beléptetőrendszerről van szó. A hash függvényünk egy iteratív hash függvény és a korábbiakban csak a tömörítőfüggvény került ismertetésre. Tárgyalásra kerül a függvény paraméterválasztása és az alkalmazott iterációs eljárás. Ezek után kerülnek megfontolásra a konkrét implementációs kérdések, a véges test kiválasztása, továbbá a futási időre és a lavinahatásra vonatkozó tesztek. Végül, mint lehetséges alkalmazási terület, a DESignIn azonosító rendszer kerül ismertetésre. A DESignIn tervezési megfontolásai közös eredményeink Huszti Andreával és Pethő Attilával és a [26] cikkben kerültek publikálásra. A szóban forgó gyakorlati vizsgálatok eredményei eddig nem lettek publikálva és a B, illetve a D Appendixben találhatóak.

A legtöbb kriptográfiai primitív esetén a szempont az, hogy gyorsan számíthatóak legyenek. A hash függvények esetén azonban az is szempont lehet, hogy a leggyorsabb implementáció is viszonylag lassú legyen. A számítási erőforrások manapság olcsók és skálázhatóak, ezért a gyors hash függvények esetén a feltöréssel próbálkozóknak is könnyebb dolga van. Thomas Roth az Amazon Elastic Compute Cloud szolgáltatásának segítségével tört fel erős kriptográfiai hash függvénnyel védett jelszavakat olcsó wifi készülékeken [1] 20 perc alatt, percenként 28 centes költséggel. Az UDHash ebből a szempontból is előnyös, hiszen a leggyorsabb implementációja is lassabb, mint a napjainkban gyakorlatban használt hash függvényeké.

A lavinahatásra vonatkozó eredmény feltételei sajnos túl szigorúak ahhoz, hogy a gyakorlati megvalósítás esetén ezeknek megfelelő paramétereket alkalmazzunk. Ezért a konstrukció lavinahatás tulajdonságára vonatkozóan gyakorlati vizsgálatok is folytak.

A lavinahatásra vonatkozó teszteredmények az B Függelékben kerülnek közlésre. Itt az egyes ábrák alatt a hash függvény paraméterezése a lineáris és a nemlineáris tagok együtthatóival, illetve a nemlineáris tag kitevőjé-

vel van megadva. A grafikon $x$ tengelyén található számok 0-761 -ig az egyes bemeneti biteket jelölik, az $y$ tengely számozása 0-243-ig a kimeneti bitekhez tartozik. Az $z$ tengely pedig 0-1499-ig a bemeneti minták számát jelöli. Egy (x,y,z) pont az ábrán azt jelenti, hogy a bemenet $x$. bitjét megváltoztatva a kimenet $y$. bitje $z$ darab minta esetében változott meg.

Látható, hogy a tesztek esetében a pontok a $z = 750$ sík környezetében találhatóak, azaz az empirikus valószínűség minden esetben a lavinahatás tulajdonságtól elvárt $\frac{1}{2}$ közelében van. Azaz, habár az UDHash a fent választott paraméterezés mellett, szigorúan vett értelemben bizonyíthatóan nem rendelkezik a lavinahatás tulajdonságával, a gyakorlatban egy, a lavinahatáshoz nagyon közeli viselkedést mutat.

A hash függvények és az álvéletlen-szám generátorok számos kriptográfiai protokollban fontos szerepet játszanak. A jelen dolgozatban ismertetett kriptográfiai primitívek lehetséges alkalmazási területének egy példája az UDSignIn azonosító rendszer.

Az UDSignIn egy univerzális webes alapú azonosítórendszernek lett tervezve. A feladata az egyszerű jelszavas bejelentkezés kiváltása, kényelmes megoldást kínál arra a problémára, hogy a biztonságos jelszavakat rendszerint nehéz megjegyezni, és hogy manapság egy átlagos felhasználó több tucat webes accounttal is rendelkezhet.

A jelszavas azonosítás esetén a biztonság érdekében a felhasználóknak minden egyes accounthoz más, erős, azaz nehezen megjegyezhető jelszót kellene észben tartania. Ez rendszerint ahhoz vezet, hogy gyenge jelszavakat használnak, vagy, hogy több helyen is ugyanazt a jelszót használják, esetleg felírják valahová. A jelszavas azonosítás felváltása tanúsítványok segítségével való azonosításra nem csak a rendszer biztonságát növeli, de a felhasználóknak is kényelmesebb megoldást jelent.

Az UDSignIn ezt a problémát egy hardvertoken segítségével hidalja át. Az azonosításhoz aszimmetrikus kriptográfiát használ, a szerver azonosításában már meglévő, széles körben elterjedt és bevált technikára, konkrétan az SSL valamelyik változatára támaszkodik. Az UDSignIn technikai leírása, és a vele kapcsolatos elméleti, gyakorlati és tervezési megfontolások a [26] cikkben kerültek publikálásra.

# 7. fejezet

# Summary

The main topics of this dissertation are two cryptographic primitives (a pseudorandom generator and a hash function) and a possible application of them.

Chapter 1 contains a short introduction. It describes cryptography as a tool of modern technology, emphasises its importance and gives an introduction to the basic notions of cryptography. After the introduction it determines the places of the primitives to study and their relation to other areas of cryptography, and it gives an overview of the whole dissertation and the main results.

The themes of Chapter 2 are the pseudorandom generators and the concept of pseudorandomness. This chapter overviews the classic pseudorandomness tests and the distinct definitions of pseudorandomness, then taking these as a starting point it reaches the pseudorandomness measures introduced by Mauduit and Sárközy ([55]). The measures in question are the Normality, the Well-distribution and the Correlation measures. The combination of the Correlation and the Well-distribution measure leads to the Combined measure ( The Normality measure is left out from the Combined measure because with the help of the Correlation measure one can give an upper bound for it). These pseudorandomness measures constitute the main tools to study the mentioned pseudorandom generator.

Chapter 3 is about the generator itself. In their paper ([55]) Sárközi and Mauduit gave an example of a pseudorandom generator having good pseudorandom measures. Later (partly with further coauthors) more good constructions were born ([39], [49] [36] [37] [56] [34]). In this chapter the common design principle of these generators: the double twist method and one of the previous generators will be described. Thereafter the construction of the new generator follows. This new generator has good pseudorandomness measures and, contrary to the previous constructions, operates over fields with even characteristics.

Also in this chapter will be proven the pseudorandomness measures of the new construction. In cryptography it is not sufficient to have pseudorandom sequences: we need large families of pseudorandom sequences. It is important to have sequences, which not only alone as a single sequence, but together as a family also have good properties. Such properties are the avalanche effect and the family complexity. Also these properties of the new construction will be proven.

The chapter discusses also the linear complexity of the generator. It turns out, that it is not suitable for cryptographic use, because its linear complexity is low. With this result a tight connection with the linear shift registers is also discovered. This connection makes possible a very fast hardware implementation, and together with the good statistical properties it makes the generator an excellent choice for non-cryptographic applications. The results of the three last sections are my results and are contained in the papers [23] and [25].

The main topic of Chapter 4 is a new hash function. The chapter starts with an introduction to hash functions in general, defines the notion of hash function and the requirements regarding the cryptographic hash functions. Thereafter the predecessor of the new construction, which was implemented under the name Codefish, will be described ([10]). The cryptanalysis of this construction (more precisely of its modified special case used by the implementation) is also part of this chapter.

Thereafter follows the new construction and the theoretical results about it. The construction fixes the flaws of its predecessor ([6]) and it means an advance regarding the operand size. Theoretical considerations constanti-

ate its preimage resistance, and one of its good statistical properties also will be proven. The investigations regarding the functions avalanche effect also take place here: although the version of the function used in the implementation does not possess the strict avalanche criterion, assymptotically it behaves very similar.

The establishment of the functions preimage resistance is a joint work with Attila Bérczes and Attila Pethő and is contained in the paper [7]. The results regarding the avalanche criterion and the theorem about the corresponding assymptotic behavior are my results and are yet unpublished.

Chapter 5 is about the implementation of the mentioned hash function and the DESignIn authentication system. Our hash function is an iterative hash function and previously only the compression function was described. The parameter selection and the iteration method will be described in this section. Thereafter the specific implementation issuses, the choice of the finite field and the test results regarding the performance and the avalanche effect will be considered. Lastly, as a possible application area, the DESignIn authentication system will be described. The design considerations of the DESignIn are joint work with Andrea Huszti and Attila Pethő and are contained in the paper [26]. The test results in question are yet unpublished and they take place in the Appendices B and D.

In the case of most cryptographic primitives it is important, that one be able to compute them fast. In the case of hash functions it can also be desirable, that the fastest implementation of the function be relatively slow. The computational resources nowadays are cheap and scalable. This means that an adversary has greater power against fast hash functions. Thomas Roth broke passwords on wifi devices protected by a strong cryptographic hash function [1]. He used the Elastic Compute Cloud service of Amazon. It took 20 minutes and costed 28 cents per minute. The UD-Hash is advantageous also from this point of view, because even its fastest implementation is slower than the nowadays widespread hash functions.

Unfortunately the conditions of the theorem regarding the avalanche effect are too strict to use parameters satisfying them in the practical implementation. Thus, there were practical investigations regarding the avalanche effect.

The test results regarding the avalanche effect are in the Appendix B. Here under the distinct figures the parametrisation is given with the coefficients of the linear and nonlinear members and the exponent of the nonlinear members. The numbers 0-761 on the $x$-axis of the diagram stand for the distinct input bits, the numbering 0-243 of the $y$-axis belongs to the output bits. The $z$-axis in turn 0-1499 denotes the amount of the input samples. An (x,y,z) point on the diagram means, that the change of the $x$. input bit changed the $y$. output bit in the case of $z$ samples.

It is easy to see, that in the case of these tests the points are in the neighbourhood of the $z = 750$ plane, that is the empiric probability is near $\frac{1}{2}$, the value expected by the avalanche effect. Thus, although the UDHash with the above described parametrisation does not possess the strict avalanche property, in practice it shows a very similar behavior to the avalanche effect.

Hash functions and pseudorandom number generators play crucial role in numerous cryptographic protocols. The UDSignIn authentication system is a possible application area for the cryptographic primitives described in this dissertation.

The UDSignIn was designed to be an universal web-based authentication system. Its task is to replace the simple password based login. It offers a comfortable solution for the problem, that the secure passwords are usually hard to memorize, and nowadays an avarage user has more dozens of account on the web.

In the case of password authentication in order to achieve secrecy the users should choose distinct strong (i.e. hard to remember) password for each of their accounts. This usually leads to the use of weak passwords or matching passwords for distinct accounts or possibly the users simply write the passwords down. The replacement of the password authentication with the help of certificates not only increases the security of the system, but also means a more comfortable solution for the users.

The UDSignIn solves this problem with the use of a hardware security token. It uses asymmetric cryptography for authentication. It also utilises the already existent and widespread SSL technology. The technical description of the UDHash and the corresponding theoretical, practical and design

considerations are contained in the paper [26].

# Irodalomjegyzék

[1] Acm tech news. http://technews.acm.org/archives.cfm?fo=2011-01-jan/jan-14-2011.html#501867.

[2] G.B. Agnew – R.C. Mullin – I.M. Onyszchuk – S.A. Vanstone: An implementation for a fast public-key cryptosystem. In *J. Cryptology*, 3. évf. (1991), 63–79. p.

[3] G.B. Agnew – R.C. Mullin – S.A. Vanstone: An implementation of elliptic curve cryptosystems over $\mathbb{F}_{2^{155}}$. In *IEEE J. Selected Areas in Comm.*, 11. évf. (1993) 5. sz., 804–813. p.

[4] Rudolf Ahlswede – Levon Khachatrian – Christian Mauduit – András Sárközy: A complexity measure for families of binary sequences. In *Period. Math. Hungar.*, 46. évf. (2003. Június) 2. sz., 107–118. p.

[5] Ágnes Andics: On the linear complexity of binary sequences. In *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.*, 48. évf. (2005), 173–180. p.

[6] Jean-Philippe Aumasson: Cryptanalysis of a hash function based on norm form equations. In *Cryptologia*, 33. évf. (2009. Január), 12–15. p. ISSN 0161-1194. URL `http://portal.acm.org/citation.cfm?id=1506456.1506458`. 4 p.

[7] A. Bérczes – J. Folláth – A. Pethő: On a family of preimage-resistant functions. In *Tatra Mountains Mathematical Publications*, 47. évf. (2010), 1–13. p.

[8] A. Bérczes – I. Járási: An application of index forms in cryptography. In *Periodica Math. Hungar.*, 58. évf. (2008), 35–45. p.

[9] A. Bérczes – J. Ködmön: Methods for the calculation of values of a norm form. In *Publ. Math. Debrecen*, 63. évf. (2003), 751–768. p.

[10] A. Bérczes – J. Ködmön – A. Pethő: A one-way function based on norm form equations. In *Periodica Mathematica Hungarica*, 49. évf. (2004), 1–13. p.

[11] E. R. Berlekamp: Factoring polynomials over large finite fields. In *Math. Comp.*, 24. évf. (1970), 713–715. p.

[12] Emile Borel: *Leçons sur la théorie des fonctions.* Gauthier-Villars, 1950. ISBN 2-87647-086-1.

[13] Nina Brandstätter – Arne Winterhof: Linear complexity profile of binary sequences with small correlation measure. In *Period. Math. Hungar.*, 52. évf. (2006. Június) 2. sz., 1–8. p.

[14] J. Buchmann – S. Paulus: A one-way function based on ideal arithmetic in number fields. In *Lect. Notes Comput. Sci.*, 1294. évf. (1997), 385–394. p.

[15] A. Cafure – G. Matera: Improved explicit estimates on the number of solutions of equations over a finite field. In *Finite Fields Appl.*, 12. évf. (2006), 155–185. p.

[16] Julien Cassaigne – Christian Mauduit – András Sárközy: On finite pseudorandom binary sequences VII: The measures of pseudorandomness. In *Acta Arith.*, 103. évf. (2002) 2. sz., 97–118. p.

[17] Çetin K.Koç – Tolga Acar: Montgomery multiplication in $GF(2^k)$. In *Designs, Codes and Cryptography*, 14. évf. (1998) 1. sz., 57–69. p.

[18] L. R. Chao – Y. C. Lin: Associative one-way function and its significances to cryptographics. In *In. J. Inf. Manage. Sci.*, 5. évf. (1994), 53–59. p.

[19] Robert Coulter – Marie Henderson: A note on the roots of trinomials over a finite field. In *Bull. Austral. Math. Soc.*, 69. évf. (2004), 429–432. p.

[20] Joan Daemen – Vincent Rijmen: The block cipher rijndael. In *Proceedings of the The International Conference on Smart Card Research and Applications* (konferenciaanyag). London, UK, 2000, Springer-Verlag, 277–284. p. ISBN 3-540-67923-5.
URL `http://portal.acm.org/citation.cfm?id=646692.759487`. 8 p.

[21] Don Davis – Ross Ihaka – Philip Fenstermacher: Cryptographic randomness from air turbulence in disk drives. In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94 konferenciasorozat. London, UK, 1994, Springer-Verlag, 114–120. p. ISBN 3-540-58333-5.
URL `http://portal.acm.org/citation.cfm?id=646759.705839`. 7 p.

[22] Whitfield Diffie – Martin Helman: New directions in cryptography. In *IEEE Transactions on Information Theory*, IT-22. évf. (1976), 644–654. p.

[23] János Folláth: Construction of pseudorandom binary sequences I. In *Period. Math. Hungar.*, 57. évf. 1. sz., 73–81. p.

[24] János Folláth: Álvéletlen számok generálása és szerepük a kriptográfiában, 2005.

[25] János Folláth: Construction of pseudorandom binary sequences II. In *Periodica Mathematica Hungarica*, 60. évf. (2010) 2. sz.

[26] János Folláth – Andrea Huszti – Attila Pethő: Asymmetric authentication system. In *Proceedings of ICAI'07 7th International Conference on Applied Informatics* (konferenciaanyag).

[27] E. Fouvry – C. Mauduit: Méthodes de crible et fonctions sommes des chiffres. In *Acta Arith.*, 77. évf. (1996) 4. sz., 339–351. p.

[28] E. Fouvry – C. Mauduit: Sommes des chiffres et nombres presque premiers. In *Math. Ann.*, 305. évf. (1996) 3. sz., 571–599. p.

[29] Róbert Freud – Edit Gyarmati: *Számelmélet.* Nemzeti Tankönyvkiadó, 2000.

[30] John Friedlander – Henryk Iwaniec: Estimates for character sums. In *Proc. Amer. Math. Soc.*, 119. évf. (1993), 365–372. p.

[31] A. O. Gelfond: Sur les nombres qui ont des propriétés additives et multiplicatives données. In *Acta Arith.*, 13. évf. (1967/1968), 259–265. p.

[32] O. Goldreich – L. Levin – N. Nisan: On constructiong 1-1 one-way functions. In *Electronic colloquium on computational complexity*, TR-95-029. évf. (1995).

[33] Oded Goldreich: *Foundations of Cryptography.* 1. köt. Cambridge Univrsity Press, 2001.

[34] Louis Goubin – Christian Mauduit – András Sárközy: Construction of large families of pseudorandom binary sequences. In *Journal of Number Theory*, 106. évf. (2004) 1. sz., 56–69. p.

[35] Great internet mersenne prime search (gimps). http://www.mersenne.org/.

[36] K. Gyarmati: On a family of pseudorandom binary sequences. In *Period. Math. Hungar.*, 49. évf. (2004), 45–63. p.

[37] K. Gyarmati: *General Theory of Information Transfer and Combinatorics*. Lecture Notes in Comp. Sci. sorozat. On a fast version of a pseudorandom generator fejezet. Springer Verlag, Heidelberg, 2006, 326–342. p.

[38] Katalin Gyarmati: Concatenation of pseudorandom binary sequences. In *Period. Math. Hungar.*, 58. évf. (2009) 1. sz., 99–120. p.

[39] Katalin Gyarmati – András Sárközy – Attila Pethő: On linear recursion and pseudorandomness. In *Acta Arith.*, 118. évf. (2005) 4. sz., 359–374. p.

[40] M.A. Hasan – M.Z. Wang – V.K. Bhargava: A modified massey-omura parallel multiplier for a class of finite fields. In *IEEE Trans. Computers*, 42. évf. (1993) 10. sz., 1278–1280. p.

[41] L. A. Hemanspaandra – J. Rothe: Creating strong, total, commutative, associative one-way function in complexity theory. In *J. Comput. Syst. Sci.*, 58. évf. (1999), 648–659. p.

[42] Tamás Herendi: *Linear Recurring Sequences*. Phd értekezés (Debreceni Egyetem). 2002.

[43] Tamás Herendi: Uniform distribution of linear recurring sequences modulo prime powers. In *Finite Fields and Their Applications*, 10. évf. (2004. Január) 1. sz., 1–23. p.

[44] Jeffrey Hoffstein – Daniel Lieman: *Cryptography and Computational Number Theory*. Progress in Computer Science and Applied Logic sorozat, 20. köt. The Distribution of the Quadratic Symbol in Function Fields and a Faster Mathematical Stream Cipher fejezet. Birkhäuser Verlag, 2001, 59–68. p.

[45] Donald E. Knuth: *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981. ISBN 0-201-03822-6.

[46] S. Lang – A. Weil: The number of points of varieties in finite fields. In *Amer. J. Math.*, 76. évf. (1954), 819–827. p.

[47] A.K. Lenstra – H.W. Lenstra Jr. – L. Lovász: Factoring polynomials with rational coefficients. In *Math. Ann.*, 261. évf. (1982), 515–534. p.

[48] Rudolf Lidl – Harald Niederreiter: *Finite Fields*. Encyclopedia of Mathematics sorozat, 20. köt. Cambridge University Press, 1997.

[49] H. Liu: A family of pseudorandom binary sequences constructed by the multiplicative inverse. In *Acta Arithmetica*, 130. évf. (2007) 2. sz., 167–180. p.

[50] Florence Jessie Collinson MacWilliams – Neil James Alexander Sloane: *The Theory of Error-Correcting Codes*. North-Holland Mathematical Library sorozat, 16. köt. North-Holland Publishing Company, 1977.

[51] J.L. Massey – J.K. Omura: Computational method and apparatus for finite field arithmetic. US Patent No. 4,587,627,, 1986.

[52] C. Mauduit – J. Rivat – A. Sárközy: Construction of pseudorandom binary sequences using additive characters. In *Monatsh. Math.*, 141. évf. (2004), 197–208. p.

[53] C. Mauduit – A. Sárközy: On the arithmetic structure of sets characterized by sum of digits properties. In *J. Number Theory*, 61. évf. (1996) 1. sz., 25–38. p.

[54] C. Mauduit – A. Sárközy: On the arithmetic structure of the integers whose sum of digits is fixed. In *Acta Arith.*, 81. évf. (1997) 2. sz., 145–173. p.

[55] Christian Mauduit – András Sárközy: On finite pseudorandom binary sequences I: Measure of pseudorandomness, the Legendre symbol. In *Acta Arith.*, 82. évf. (1997) 4. sz., 365–377. p.

[56] Christian Mauduit – András Sárközy: Construction of pseudorandom binary sequences by using the multiplicative inverse. In *Acta Math. Hungar.*, 108. évf. (2005) 3. sz., 239–252. p.

[57] Alfred J. Menezes – Paul C. van Oorschot – Scott A. Vanstone: *Handbook of Applied Cryptography*. CRC Press, 2001.

[58] R. C. Merkle: A fast software one-way hash function. In *J. Cryptology*, 3. évf. (1990), 43–48. p.

[59] Ivan Niven – H. S. Zuckerman: On the definition of normal numbers. In *Pacific J. Math.*, 1. évf. (1951) 1. sz., 103–109. p.

[60] G. I. Perelmuter: On certain character sums. In *Uspehi Mat. Nauk*, 18. évf. (1963) 2. sz., 145–149. p.

[61] Arash ReyhaniMasoleh – Anwar Hasan: Fast normal basis multiplication general purpose processors. In *IEEE Transactions on Computers*, 52. évf. (2003) 11. sz., 1379–1390. p.

[62] R. L. Rivest – A. Shamir – L. Adleman: A method for obtaining digital signatures and public-key cryptosystems. In *Commun. ACM*, 21. évf. (1978. Február), 120–126. p. ISSN 0001-0782.
URL `http://doi.acm.org/10.1145/359340.359342`. 7 p.

[63] András Sárközy: A finite pseudorandom binary sequence. In *Studia Sci. Math. Hungar.*, 38. évf. (2001), 377–384. p.

[64] Wolfgang Schmidt: A lower bound for the number of solutions of equations over finite fields. In *J. Number Theory*, 6. évf. (1974), 448–480. p.

[65] Wolfgang Schmidt: *Equations over Finite Fields. An Elementary Approach*. Lecture Notes in Mathematics sorozat, 536. köt. Springer-Verlag, 1976.

[66] Secure socket layer (ssl). http://wp.netscape.com/eng/ssl3/.

[67] I. Shparlinski: *Number theoretic methods in cryptography.* Progress in Computer Science and Applied Logic sorozat, 17. köt. Birkhäuser Verlag, Basel, 1999.

[68] Douglas R. Stinson: *Cryptography - Theory and Practice.* Discrete Mathematics and its Applications sorozat. Chapman & Hall/CRC, 2006.

[69] Q. Sun: A kind of trap-door one-way function over algebraic integers. In *J. Sichuan Univ., Nat. Sci. Ed.*, 2. évf. (1986), 22–27. p.

[70] B. Sunar – C.K. Koc: An efficient optimal normal basis type ii multiplier. In *IEEE Trans. Computers*, 50. évf. (2001) 1. sz., 83–88. p.

[71] Aimo Tietäväinen: *Algebra, some current trends.* Lecture Notes in Math. sorozat, 1352. köt. Incomplete sums and two applications of Deligne's result fejezet. Springer Verlag, New York, 1988, 190–205. p.

[72] Viktória Tóth: Collision and avalanche effect in families of pseudorandom binary sequences. In *Period. Math. Hungar.*, 55. évf. (2007. November) 2. sz., 185–196. p.

[73] Ivan Matveyevich Vinogradov: *Elements of Number Theory.* Dover Publications, 2003.

[74] Joachim von zur Gathen – Jürgen Gerhard: *Modern Computer Algebra.* Cambridge University Press, 1999.

# A. függelék

# Tesztkonfiguráció

## A.1. Hardver

```
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Celeron(R) CPU        E3400  @ 2.60GHz
stepping        : 10
cpu MHz         : 2600.947
cache size      : 1024 KB
cpu cores       : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
                  pge mca cmov pat pse36 clflush dts acpi mmx fxsr
                  sse sse2 ss ht tm pbe syscall nx lm constant_tsc
                  arch_perfmon pebs bts rep_good nopl aperfmperf
                  pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16
                  xtpr pdcm xsave lahf_lm dts tpr_shadow vnmi
```

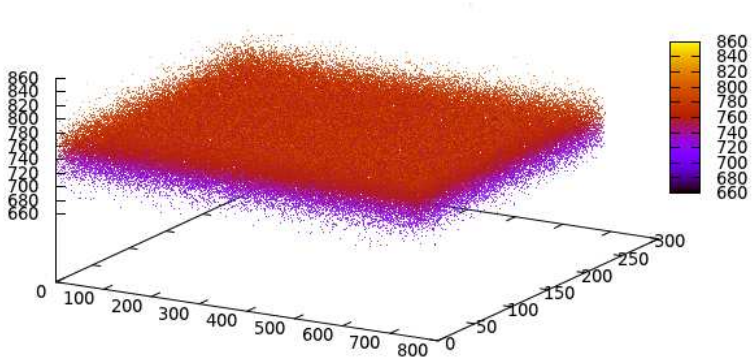
```
                  flexpriority
bogomips        : 5201.89
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
MemTotal        : 2021360 kB
```

## A.2. Szoftver

| Szoftver | Verzió |
|---|---|
| Ubuntu | 11.04 |
| gcc | 4.5.2 |
| GNU MP | 4.3.2 |
| Crypto++ | 5.6.0 |

# B. függelék

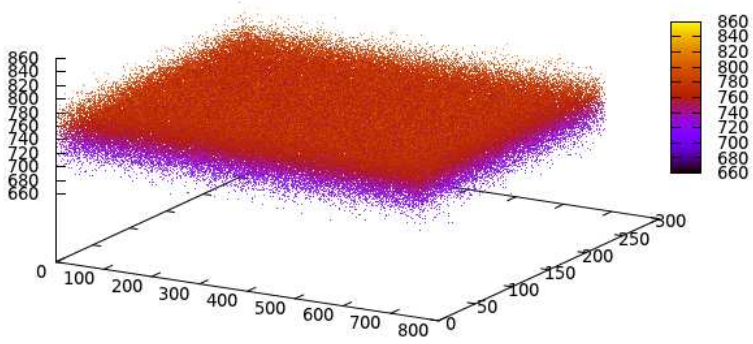
# Lavinahatás tesztek




B.1. ábra. Alap teszteredmények

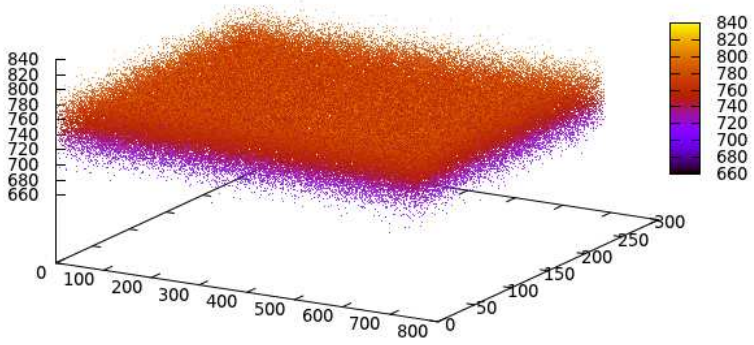
**Kitevő:** 286295073
**Az összetett tag együtthatói:**
3d1b58ba507ed7ab625558ec238e1f2974b0dc5119495cff327b23c6643c9869,

519b500d431bd7b76b68079a4e6afb667fdcc2331befd79f3352255a109cf92e,
3a95f874081386412ca886110836c40e189a769b54e49eb479838cb24353d0cd
**A lineáris tag együtthatói:**
1190cde766ef438d4db127f80216231b515f007c5bd062c241b71efb79e2a9e3,
6763845e75a2a8d4721da3172443a858436c6125628c895d7c83e458257130a3,
440badfc05072367614fd4a1419ac24122221a704516dde97c3dbd3d737b8ddc



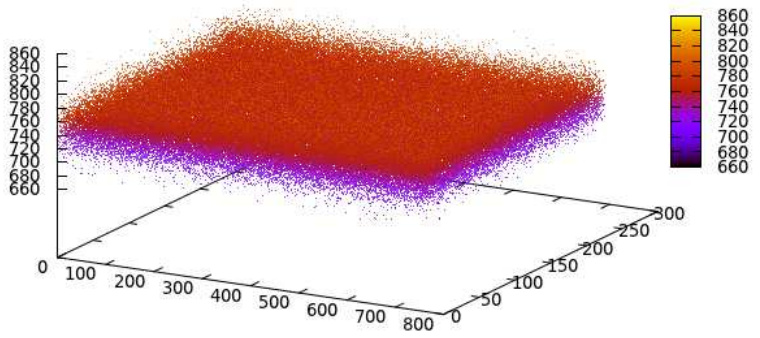

B.2. ábra. Teszteredmények más együtthatókkal

**Kitevő:** 286295073
**Az összetett tag együtthatói:**
680c1b62443d8e597a1c574377d02f4866613afa24a7c25e0eed54217fcf7bf2,
6d5e56326881ca2d1642e53a02208f331e81e2ee73333b123d5c5c2d37cd332f,
2df5b11c2fdc7fd51ad7b7c6115a9ce6651e5e3b53fe40b97e8820947c2e42da
**A lineáris tag együtthatói:**
57cffdde091852a979ef5e9e462d4a4336cc7f001ad4086200c828696a5674f1,
1c1463c6023ee43c7e674f6b4b72653c27954eaa4aaa3cd33c9179df629e2171,
6453d5b017a483cb6a9868e9700f352168ad2b6a337dbb410b69b04a461f650f




B.3. ábra. Teszteredmények kis kitevővel

**Kitevő:** 127
**Az összetett tag együtthatói:**
3d1b58ba507ed7ab625558ec238e1f2974b0dc5119495cff327b23c6643c9869,
519b500d431bd7b76b68079a4e6afb667fdcc2331befd79f3352255a109cf92e,
3a95f874081386412ca886110836c40e189a769b54e49eb479838cb24353d0cd
**A lineáris tag együtthatói:**
1190cde766ef438d4db127f80216231b515f007c5bd062c241b71efb79e2a9e3,
6763845e75a2a8d4721da3172443a858436c6125628c895d7c83e458257130a3,
440badfc05072367614fd4a1419ac24122221a704516dde97c3dbd3d737b8ddc




B.4. ábra. Teszteredmények alacsony súlyú kitevővel

**Kitevő:** 268435459 ($2^{28} + 2^1 + 2^0$)
**Az összetett tag együtthatói:**
3d1b58ba507ed7ab625558ec238e1f2974b0dc5119495cff327b23c6643c9869,
519b500d431bd7b76b68079a4e6afb667fdcc2331befd79f3352255a109cf92e,
3a95f874081386412ca886110836c40e189a769b54e49eb479838cb24353d0cd
**A lineáris tag együtthatói:**
1190cde766ef438d4db127f80216231b515f007c5bd062c241b71efb79e2a9e3,
6763845e75a2a8d4721da3172443a858436c6125628c895d7c83e458257130a3,
440badfc05072367614fd4a1419ac24122221a704516dde97c3dbd3d737b8ddc

# C. függelék

# Sebességtesztek UDHash függvényei

## C.1. UDHash254

**Kitevő:**

12

**Az összetett tag együtthatói:**

3d1b58ba507ed7ab625558ec238e1f2974b0dc5119495cff327b23c6643c9869,
519b500d431bd7b76b68079a4e6afb667fdcc2331befd79f3352255a109cf92e,
3a95f874081386412ca886110836c40e189a769b54e49eb479838cb24353d0cd

**A lineáris tag együtthatói:**

1190cde766ef438d4db127f80216231b515f007c5bd062c241b71efb79e2a9e3,
6763845e75a2a8d4721da3172443a858436c6125628c895d7c83e458257130a3,
440badfc05072367614fd4a1419ac24122221a704516dde97c3dbd3d737b8ddc

## C.2. UDHash509

**Kitevő:**

12

**Az összetett tag együtthatói:**

1190cde766ef438d4db127f80216231b515f007c5bd062c241b71efb79e2a9e33
d1b58ba507ed7ab625558ec238e1f2974b0dc5119495cff327b23c6643c9869,
440badfc05072367614fd4a1419ac24122221a704516dde97c3dbd3d737b8ddc3
a95f874081386412ca886110836c40e189a769b54e49eb479838cb24353d0cd,
15b5af5c741226bb235ba86147398c89393865751cf10fd823f9c13c649bb77c5
20eedd1374a3fe675c6c33a12e685fb3dc240fb1ba026fa684a481a579478fe

**A lineáris tag együtthatói:**

6763845e75a2a8d4721da3172443a858436c6125628c895d7c83e458257130a35
19b500d431bd7b76b68079a4e6afb667fdcc2331befd79f3352255a109cf92e,
38437fdb7644a45c4b588f54542289ec2cd89a3257e4ccaf2a487cb01d4ed43b3
855585c70a64e2a2d517796580bd78f2463b9ea5e884adc77465f017724c67e,
06b9476442c296bd5f5e7fd0098a3148100f8fca6590700b1f48eaa11381823a2
5a70bf71dbabf00310c50b35ff87e057e0c57b177ae35eb10233c993f6ab60f

## C.3. UDHash256

**Modulus:**

628503907844138419186562739485275657779508862386893721474692236956
01627048263

**Kitevő:**

31378397166559283966765678011653532519 3

**Az összetett tag együtthatói:**

616309944986795191724102638347419151468155942153823823366260288647932794506 18,
592023207485630449627802331363236261768565619144303109483719501936832900732 01,
188959985259665703319191147102059767066184752960342748287576331774770187171 00

**A lineáris tag együtthatói:**

188013062533933815456187202687414331943249015883608319900568154091737082579 48,
492050709691232541497239841724811202778208114412962566757854714927771620312 59,
266899935942294966963389662896300056882392916297827041542384623713215220208 64

## C.4. UDHash512

**Modulus:**

133478789543240246517116323734764591884281378818183507238665895295526493340607964443944184880168629528021893178493029621597809928973535217825503222662722 81

**Kitevő:**

111014492188910947555385041036688199156943917908453052200992005575727851660423

**Az összetett tag együtthatói:**

132676712675572896158444304737378609350249409211650836653473960506423942152711717821090055542067039261190771708076263565960324816071022659974703106215015 56,
103872745013076925645389271273070726734975631558612552837500952206995956662303038885160563331097994241537529525195701833473734458710

652713374387269576291 58,
856926752093821826249200817054868331971866442448138071078954740205
864045948664742215074151408487318632640247217873934926478637674330
831735032505379799084 7

**A lineáris tag együtthatói:**

166023202398036249765559080093058335012423535990578413377113805080
951596890276819993045743652117163294204753520471263301114058996729
2643616109292525964756,
310017269523333243359762637274163049508884541180672327806590158988
085273256214451024050621432497214755406010180048968763682391327973
894819714050552155859 9,
123733747304399180791823208711197904759064144114306956412667681673
132280085439577466844902890978619144292692380712862285347513780623
3630487705232261882845 9

# D. függelék

# Sebességtesztek

| Hash | Megabájt/másodperc |
|---|---|
| UDHash254 (páros karakterisztika) | 0.019 |
| UDHash509 (páros karakterisztika) | 0.017 |
| UDHash256 (páratlan karakterisztika) | 1.06 |
| UDHash512 (páratlan karakterisztika) | 0.44 |
| CRC32 | 441 |
| Adler32 | 1191 |
| MD5 | 420 |
| SHA-1 | 223 |
| SHA-256 | 160 |
| SHA-512 | 197 |
| Tiger | 356 |
| Whirpool | 85 |
| RIPEMD-160 | 171 |
| RIPEMD-320 | 186 |
| RIPEMD-128 | 256 |
| RIPEMD-256 | 278 |

# E. függelék

# Publikációs jegyzék

## E.1. Publikációk

1. J. Folláth, A. Huszti, A. Pethő, DeSignIn Asymmetric Authentication System, Proceedings of ICAI'07 7th International Conference on Applied Informatics, pp. 53-61.

2. J. Folláth, Construction of Pseudorandom Binary Sequences Using Additive Characters Over GF($2^k$), Periodica Mathematica Hungarica Vol. 57 (1), 2008, pp. 73-81.

3. A. Bérczes, J. Folláth, A. Pethő, On a Family of Preimage-Resistant Functions, Tatra Mountains Mathematical Publications, 47, 1-13 (2010)

4. J. Folláth, Construction of Pseudorandom Binary Sequences Using Additive Characters Over GF($2^k$) II., Periodica Mathematica Hungarica, Volume 60, Number 2, 2010, pp. 127-135

5. L. Aszalós, N. Bátfai, L. Csirmaz, J. Folláth, T. Herendi, T. Kovács, A. Pethő, P. Varga, Secure Utilisation of Local and Regional Data Assets through Mobile Environments, Proceedings

of ICAI’10 8th International Conference on Applied Informatics, beküldve

6. J. Folláth, A. Huszti, A. Pethő, Informatikai biztonság és kriptográfia, Kempelen Farkas Digitális Tankönyvtár, megjelenés alatt

## E.2. Szoftverek

- DESignIn bejelentkező rendszer
  http://www.inf.unideb.hu/designin/signature/login.php
- A 2. cikkben ismertetett álvéletlenszám generátor implementációja
  http://www.inf.unideb.hu/~follathj/downloads/gf2k-dev_1.0.tar.gz
- A 3. cikkben ismertetett hash függvény implementációja
  http://www.inf.unideb.hu/~follathj/downloads/udhash.tar.gz