Protocols criptogràfics basats en dades biomètriques

MEM-Processament, transport i protecció de la informació

Gemma Casaña Codina
Director Carles Padró Laimon

Octubre 2008

Índex

Introducció

En aquest treball s'estudien els protocols criptogràfics usats per tractar dades biomètriques i les possibles millores per a aquests protocols. Un senyal biomètric es pot definir com una característica física o psicològica única, mesurable i biològica que permeti reconèixer o verificar la identitat d'una persona. Podem classificar els senyals biomètrics en dos tipus tal i com es fa a [20]: els estàtics (empremtes dactilars, iris, retina,...) i els dinàmics basats en el comportament de l'usuari en desenvolupar certes tasques (reconeixement de veu, manera de caminar, firma manuscrita,...).

El sistema més usat és el de les empremtes dactilars (basat normalment en l'estudi dels punts singulars de l'empremta anomenats minúcies) degut a les seves propietats d'unicitat i permanència i a la seva relació fiabilitat/preu (vegeu [25]). Tanmateix el sistema que reporta resultats més acurats actualment és l'escàner d'iris però els equips són bastant cars i voluminosos, fet que els fa poc pràctics (per a més informació sobre l'escàner d'iris consulteu [15]).

Els sistemes actuals que usen senyals biomètrics no creen un lligam veritable entre el senyal biomètric i la clau criptogràfica. De fet, aquest lligam és simplement una resposta binària d'un verificador, fet que disminueix la seguretat considerablement. És per això que s'hauran d'usar conjuntament la biometria i la criptografia. La biometria aboleix el problema que tenen molts sistemes criptogràfics actuals respecte la falta d'autentificació de l'usuari, és a dir, no hi ha cap garantia que la clau estigui sent introduïda per l'usuari legítim (sobretot en connexions informàtiques). A més a més, també comporta molts avantatges a l'hora de tractar amb claus criptogràfiques grans ja que permet que l'usuari disposi d'una clau suficientment llarga sense haver-la de recordar i elimina la utilització de PINs o contrasenyes fàcils de recordar que fan el sistema molt més vulnerable.

És per tot això que la biometria està sent objecte d'estudi de molts investigadors i està començant a ser utilitzada no solament per grans empreses sinó també per particulars acostant aquests sistemes biomètrics a qualsevol usuari (vegeu per exemple [16], [17], [18] i [19]). Malgrat això, aquest ús que comença a ser globalitzat planteja forces qüestions no solament èticomorals (és el cas de les grans bases de dades biomètriques per part d'organismes policials o governamentals) sinó també qüestions referents a la seguretat i

la difusió controlada d'aquestes dades tal i com s'expressa a [20]. És per això que és molt important obtenir protocols criptogràfics per poder tractar aquestes dades amb la seguretat necessària ja que, si ens roben un PIN sempre se'n pot regenerar un de nou, fet impensable amb un senyal biomètric com una empremta dactilar.

Les dades biomètriques presenten diversos problemes que impedeixen que puguin ser tractades pels sistemes criptogràfics actuals, com per exemple el fet que dos escanejats d'un mateix senyal biomètric no seran mai idèntics. Amb l'objectiu d'usar les dades biomètriques per obtenir claus criptogràfiques segures (per xifrar i desxifrar missatges o per controlar l'accés a certs espais) s'han desenvolupat diferents protocols dels quals n'estudiarem dos tipus: la *Biometric encryption* i els *fuzzy extractors*. Un cop presentats aquests dos sistemes hem fet la nostra contribució realitzant un estudi de les possibles millores per als dos sistemes intentant aportar les propietats que els manquen. Finalment, hem realitzat l'estudi d'una part més matemàtica dedicada als *codis detectors de manipulacions algebraiques* (AMD-*codes*) que ens serviran com a eina per poder introduir robustesa en els dos tipus de sistemes biomètrics estudiats.

Així doncs, després de fer una introducció al món de la biometria en el Capítol 1 passem a l'estudi dels *fuzzy extractors* i de la *Biometric encryption* en els Capítols 2 i 3 respectivament. Després d'aquesta presentació dels sistemes passem al Capítol 4 on aportem un estudi de possibles millores per a ambdós sistemes. En alguns casos modifiquem el sistema inicial per implementar la millora, mentre que en d'altres casos només contribuïm amb un estudi dels possibles problemes que ens podríem trobar si volguéssim dur a terme la millora. Finalment en el Capítol 5 trobem la part més matemàtica del treball amb l'estudi dels AMD-*codes* que inclouen propietats i teoremes de teoria de codis, combinatòria i criptografia i que fonamenten les propietats de robustesa en els dos protocols estudiats.

El primer dels sistemes estudiats són els *fuzzy extractors* en el Capítol 2, aquest sistema es basa en dues fases:

- Durant la primera fase l'usuari introdueix el seu input ω (que serà el seu senyal biomètric o certa informació que es pugui derivar d'aquest) i el *fuzzy extractor* ens retorna la clau privada R i la seqüència pública d'ajuda P. La clau privada es farà servir i posteriorment es descartarà conservant únicament la seqüència pública d'ajuda P.

- Durant la segona fase l'usuari voldrà tornar a fer servir la clau privada R i, per tant, l'haurà de regenerar. Per això introduirà el seu input ω' (potser lleugerament modificat) juntament amb P i el *fuzzy extractor* ens retornarà la clau R inicial.

El concepte de *fuzzy extractor* es va introduir en [5] però en aquesta versió les comunicacions s'havien de fer en un canal autentificat, és a dir no hi havia garanties de seguretat

quan l'adversari modificava P. Posteriorment en [4] s'introdueix la noció de *fuzzy extractor* robust que soluciona aquest problema però, en aquesta primera aproximació als *fuzzy extractors* robustos, la clau extreta era significativament més curta que en el cas no robust. Finalment, [6] introdueix una nova versió de *fuzzy extractors* robustos on la clau extreta és prou llarga gràcies a la introducció dels paràmetres públics del sistema. No obstant, recentment s'ha presentat una nova versió de *fuzzy extractor* robust en [7] que no utilitza paràmetres públics del sistema i que també millora considerablement els resultats de llargada de la clau obtinguts en [4].

Els *fuzzy extractors* usen tres procediments auxiliars que s'estudiaran amb deteniment. Aquests són:

- Els *randomness extractors* que, com el seu nom indica, ens permetran obtenir dades uniformement aleatòries a partir d'altres que no ho són (o no tenen perquè ser-ho).
- Els *secure sketches* que ens permetran recuperar qualsevol seqüència $\omega \in \mathcal{M}$, on $\mathcal{M}$ és l'espai mètric on treballem, a partir d'un $\omega' \in \mathcal{M}$ prou proper a ω (segons la distància que considerem en $\mathcal{M}$) difonent la mínima informació d'ω possible.
- Els codis d'autentificació de missatges (MAC) que evitaran la possible manipulació de les dades públiques per part d'algun adversari. Més concretament, si aquesta modificació es duu a terme ens n'adonarem amb probabilitat alta. En particular per a la construcció dels *fuzzy extractors* usarem MACs amb seguretat contra manipulacions de clau, és a dir KMS-MACs.

Els *fuzzy extractors* tenen l'avantatge principal que incorporen la propietat de robustesa, però en canvi no tenen fase de tractament de la imatge, fet que el sistema de la *Biometric encryption* estudiat amb deteniment en el Capítol 3 sí que incorpora.

Aquest segon sistema va ser desenvolupat per Mytec Technologies Inc. (vegeu [2] i [13]). La idea és la mateixa que la usada en els *fuzzy extractors*, és a dir, usar un senyal biomètric per poder xifrar i desxifrar missatges de forma segura i sense haver de recordar cap PIN o contrasenya. No obstant això, el procediment que fa servir es força diferent.

El sistema de la *Biometric encryption* consta, igual que els *fuzzy extractors*, de dues fases:

- Durant la primera fase l'usuari introdueix el seu input x i el sistema ens retorna la clau privada k_0 i una informació pública anomenada *Bioscrypt*. La clau privada es farà servir i posteriorment es descartarà conservant únicament la informació pública.
- Durant la segona fase l'usuari voldrà tornar a fer servir la clau privada k_0 i, per tant, l'haurà de regenerar. Per això introduirà el seu input x' (potser lleugerament modificat) juntament amb el *Bioscrypt* i el sistema ens retornarà la clau k_0 inicial.

Per dur a terme aquestes fases utilitzarem tres procediments bàsics:

- El procediment per al tractament de la imatge consistirà en extreure un filtre a partir de les dades inicials durant la primera fase per filtrar tant aquestes dades com les que obtindrem en la segona fase (que possiblement estaran distorsionades).
- El *link algorithm* ens servirà per lligar la clau criptogràfica (que es genera aleatòriament) amb el senyal biomètric filtrat. Per fer-ho es crearà una taula $\mathbf{M}$ que s'emmagatzemarà en el *Bioscrypt* de tal manera que, només amb aquesta taula no es podrà recuperar ni el senyal biomètric ni la clau criptogràfica però, juntament amb un nou escanejat del mateix senyal es podrà regenerar la clau k_0.
- L'identificador serà usat per garantir que la clau regenerada en la segona fase és la mateixa que la inicial.

Com ja hem explicat, la *Biometric encryption* té el problema de la falta de robustesa fet que la fa més vulnerable que els *fuzzy extractors* en temes de seguretat contra alguns atacs (vegeu [12]). Un altre dels inconvenients de la *Biometric encryption* és la falta d'un lligam tan fort entre el senyal biomètric i la clau criptogràfica com el que s'aconsegueix en els *fuzzy extractors*. En la *Biometric encryption* la clau és triada aleatòriament i posteriorment unida amb el senyal biomètric. En canvi, en els *fuzzy extractors* la clau criptogràfica és extreta directament del senyal biomètric. No obstant, aquesta propietat comporta l'inconvenient de la limitació de llargada de la clau en els *fuzzy extractors*.

Vistos aquests dos sistemes, hem intentat aconseguir alguna millora per a aquests sistemes estudiant diverses possibilitats. En el Capítol 4 s'estudien possibles millores per a ambdós sistemes. En particular, per als *fuzzy extractors* s'estudia la possibilitat d'introduir una fase de tractament de la imatge, però sense massa èxit ja que aquest fet comporta molts problemes de coherència de sistemes mètrics. A més a més, també s'estudia el problema de la recuperació d'ω en la segona fase (la de verificació) i la possible pèrdua de seguretat que comporta aquest fet, però tampoc s'arriba a cap resultat positiu. Per al sistema de la *Biometric encryption* s'aconsegueixen millors resultats. En primer lloc aquest sistema, en particular el *link algorithm* usa un codi de repetició. En aquest capítol de millores s'ha realitzat un estudi amb diferents codis (extrets de [10]) per intentar trobar-ne un de millor. La resposta no és tan òbvia com podria semblar en un principi perquè el codi de repetició té una capacitat correctora molt baixa però, si els errors estan disposats de manera adequada en pot arribar a corregir aproximadament un 40% de la informació. Podem millorar la capacitat correctora mantenint el mateix percentatge amb algunes combinacions de codis (Hamming + Repetició o Golay + Repetició) no obstant podem aconseguir una capacitat correctora molt més elevada (que ens asseguraria la correcció

d'errors en qualsevol disposició) amb un codi de Reed-Muller $[512, 46, 128]$ fent servir 1536 bits i disminuint el percentatge a un $12,3\%$. En segon lloc també estudiem la possibilitat d'introduir robustesa al sistema de la *Biometric encryption* fent servir les mateixes eines que es fan servir per aconseguir robustesa en els *fuzzy extractors*, és a dir, substituir la fase de l'identificador per un KMS-MAC que controli les possibles modificacions del *Bioscrypt* per part d'algun adversari. D'aquesta manera es solucionaria la falta de robustesa en aquest sistema.

Finalment en aquest capítol de millores també es plantegen els problemes que sorgeixen a l'hora de crear un híbrid entre els dos sistemes és a dir, prendre la fase de tractament de la imatge de la *Biometric encryption* i fer-la servir en un *fuzzy extractor* robust. De totes maneres no s'arriba a cap conclusió ferma, perquè hi ha bastants problemes auxiliars com la falta de coherència de sistemes mètrics o la falta d'informació sobre l'entropia que presenten les dades biomètriques, entre d'altres.

Finalment en el Capítol 5 es presenta un estudi dels codis detectors de manipulacions algebraiques (AMD-*codes*) necessaris per construir els KMS-MAC, usats per introduir robustesa en els dos sistemes que s'han vist. Un (S, G, δ) AMD-*code* és una funció probabilística de codificació $\mathcal{E} : \mathcal{S} \to \mathcal{G}$ d'un conjunt $\mathcal{S}$ de mida S en un grup $\mathcal{G}$ d'ordre G juntament amb una funció per descodificar $\mathcal{D} : \mathcal{G} \to \mathcal{S} \cup \{\perp\}$ tal que $\mathcal{D}(\mathcal{E}(s)) = s$ amb probabilitat 1 per a qualsevol $s \in \mathcal{S}$. També demanarem que per a qualsevol $\Delta \in \mathcal{G}$:

$$\Pr[\mathcal{D}(\mathcal{E}(s) + \Delta) \notin \{s, \perp\}] \leq \delta$$

En els (S, G, δ) AMD-*codes* amb seguretat dèbil, el paràmetre s ha d'estar triat de manera uniforme i aleatòria però en els AMD-*codes* amb seguretat forta la condició es compleix per a qualsevol $s \in \mathcal{S}$ és a dir, que s podria haver estat triat de manera no aleatòria (per exemple un adversari podria haver triat aquest paràmetre maliciosament). Els AMD-*codes* més comuns són els sistemàtics caracteritzats per tenir la funció per codificar amb la forma següent:

$$\begin{aligned} \mathcal{E} : \ \mathcal{S} \ &\to \ \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \\ s \ &\mapsto \ (s, x, f(x, s)) \end{aligned}$$

per a alguna funció f i $x \in_R \mathcal{G}_1$. La descodificació ve donada per:

$$\mathcal{D}(\tilde{s}, \tilde{x}, \tilde{\sigma}) = \begin{cases} \tilde{s} & \text{si } \tilde{\sigma} = f(\tilde{x}, \tilde{s}) \\ \perp & \text{altrament} \end{cases}$$

També en aquest capítol veurem la relació entre els AMD-*codes* i la combinatòria, així com com les seves aplicacions a l'hora de construir AMD-*codes* amb seguretat forta a partir d'un MAC i un AMD-*code* amb seguretat dèbil i construir KMS-MACs a partir d'AMD*codes* amb seguretat forta.

A continuació es revisen algunes de les primeres construccions d'AMD-*codes* fetes per Ogata i Kurosawa [9], per Cabello, Padró i Sáez [1]. Aquestes construccions presenten el problema de no ser escalables és a dir, tot i que s'acosten a les fites proposades la *tag size* efectiva s'allunya molt de l'òptim i per tant no es pot augmentar tant com vulguem la mida del conjunt de sortida. També en [9] es presenta una versió d'AMD-*code* amb seguretat dèbil que sí que és escalable, és a dir, per a una mida constant de redundància podem augmentar tant com vulguem la mida del conjunt de sortida. Tanmateix per als AMD-*codes* amb seguretat forta s'han trobat fites que descarten la possibilitat de poder trobar construccions totalment escalables. De totes maneres s'han aconseguit AMD-*codes* amb seguretat forta molt flexibles.

Una de les construccions genèriques més recents fan servir codis correctors d'errors. Per a què les construccions siguin efectives s'ha de treballar amb paràmetres dels codis diferents als normals. Usant un codi de Reed-Solomon lleugerament modificat s'arriba a una de les construccions més recents proposada a [3]. Aquesta aconsegueix una *tag size* efectiva molt propera a l'òptim però si es vol augmentar la mida del conjunt de sortida s'ha de canviar tota l'estructura de l'AMD-*code*. Finalment es presenta una construcció que utilitza l'AMD-*code* amb seguretat dèbil escalable presentat en [9] i una família de codis correctors d'errors per crear un AMD-*code* amb seguretat forta molt més flexible que les anteriors. A més a més la *tag size* efectiva tampoc difereix molt de l'òptim i els codis no necessiten complir unes propietats tan exigents com en les construccions anteriors.

Capítol 1

Biometria

1.1 Introducció a la biometria

Un senyal biomètric es pot definir com una característica física o psicològica única, mesurable i biològica que permeti reconèixer o verificar la identitat d'una persona.

L'autentificació biomètrica ha estat practicada des de sempre per animals i éssers humans: reconèixer una veu quan responem al telèfon implica la posada en funcionament d'un complex mecanisme que compara la veu que sentim amb totes les que tenim emmagatzemades al nostre cervell. També trobem exemples en alguns animals, que es poden reconèixer a través de senyals biomètrics com l'olor corporal.

Actualment s'estan usant sistemes biomètrics en empreses, organitzacions i governs de tot el món per al control d'accés tant físic com virtual (xarxes, bases de dades, transaccions electròniques,...).

No fa molts anys els sistemes biomètrics (com els lectors d'empremtes dactilars o els d'iris) s'usaven només en edificis governamentals o en grans empreses, però actualment moltes biblioteques, clubs esportius, supermercats i fins i tot usuaris privats usen la biometria per agilitzar i disminuir costos en les tasques d'identificació de personal.

Els sistemes actuals de verificació biomètrica es poden separar en dos tipus tal i com es fa a [20], la biometria estàtica i la dinàmica. En el primer apartat trobem el mesurament de característiques físiques i estàtiques de l'individu, com per exemple les empremtes dactilars o l'iris. En el segon apartat trobem el mesurament de paràmetres dinàmics de l'usuari basats en el seu comportament en desenvolupar certes tasques. En la Taula 1.1 trobem alguns dels principals senyals biomètrics que es fan servir actualment.

El sistema més usat actualment és el de lectura d'empremtes dactilars (vegeu [25]). Això és degut a les seves qualitats d'unicitat (la probabilitat de coincidència entre dues persones ha de ser el més baixa possible) i permanència (ha de ser poc susceptible al pas del temps i als canvis en un individu).

Biometria estàtica	**Biometria dinàmica**
Empremta dactilar	Reconeixement de la veu
Iris	Manera de caminar
Retina	Dinàmica de teclejar
Veu	Firma manuscrita
Forma de la cara	Anàlisi gestual
Venes de la mà	...
Forma de l'orella	
ADN	
Olor corporal	
Termografia	
Geometria de la mà	
...	

Taula 1.1: Classificació dels principals senyals biomètrics actuals

Avui en dia el mercat està ple de lectors d'empremtes digitals assequibles tant per a empreses com per a usuaris privats. És un sistema ràpid i senzill d'usar amb una molt bona relació fiabilitat-preu (vegeu [16], [17], [18] i [19] d'exemples).

Malgrat això, el sistema que reporta resultats més acurats és l'escàner de l'iris, però actualment els equips són bastant cars i voluminosos, fet que els fa poc pràctics en nombroses situacions (per a més informació sobre l'escàner d'iris consulteu [15]).

Un dels sistemes típics és el reconeixement de la veu, però és poc fiable degut a que la veu humana està subjecte a canvis externs per malaltia, afonia o altres problemes vocals. Aquest sistema també és força vulnerable a sorolls externs o problemes de comunicació.

Les vies actuals d'investigació es mouen en diferents direccions. Per una banda s'intenten millorar els sensors actuals per aconseguir uns resultats més precisos. Per altra banda també es vol millorar la seguretat dels sistemes actuals i fer-los menys vulnerables als possibles atacs. També s'estudia la possibilitat d'implementar la biometria multimodal, és a dir, combinar diferents tipus de senyals biomètrics (com per exemple reconeixement de la veu, forma de la cara i lectura d'empremtes digitals) per augmentar la fiabilitat de l'autentificació. Finalment també s'està estudiant la possibilitat de crear targetes intel·ligents que permetin emmagatzemar i transportar les dades biomètriques personals (vegeu [20]).

De fet, la recent implementació del DNI digital es podria incloure dins d'aquesta última via d'estudi. Aquest document que podem veure en la Figura 1.1, incorpora un xip que no tan sols conté les mateixes dades impreses en el DNI sinó que també inclou un patró de les nostres empremtes dactilars, les imatges digitalitzades de la nostra firma i de la nostra imatge en blanc i negre i també dos certificats que permeten firmar i autentificar

electrònicament documents per tal d'acreditar la identitat de l'usuari quan no sigui possible el reconeixement físic. Aquestes dades estan protegides amb un PIN que només coneix el propietari del DNI (per a més informació vegeu [27]).

Figura 1.1: DNI electrònic

1.2 Empremtes dactilars

Com ja hem explicat en la secció anterior, les empremtes dactilars són el mètode més usat actualment. L'empremta dactilar està formada per un seguit de crestes i valls de l'epidermis de la punta del dits que generalment segueixen línies paral·leles. Aquestes línies es corben, s'uneixen les unes amb les altres, es tallen bruscament, es bifurquen... La identificació per empremtes dactilars es fa basant-se en aquests punts singulars que podem observar en la Figura 1.2. Dels punts on s'acaba una línia o d'on es bifurca una línia se'n diuen minúcies i formen pràcticament el 80% dels punts singulars de l'empremta. De cada minúcia se'n diu la posició en coordenades i la inclinació i això permet quantificar l'empremta. Per a més informació sobre aquest tema vegeu [14], [21] i [26].

Figura 1.2: Empremta dactilar i minúcies més comunes

Cap als anys 30, quan es van començar a usar bases de dades d'empremtes dactilars per a la identificació de criminals, aquestes cerques es feien manualment. Cap al 1960 l'FBI va començar a automatitzar el procés, i en aquesta època també van començar a desenvolupar-se sistemes de control d'accés (vegeu [24]).

Cal remarcar que, fins al moment no està científicament demostrat quin és el grau de fiabilitat en la identificació per empremtes dactilars tal i com es comenta a [22]. A diferència d'altres mètodes com el de la identificació per ADN que té ratios d'error conegudes, en el cas de les empremtes diversos estudis donen ratios que oscil·len entre un $0,8\%$ i un 4% d'error. És a dir, que aquesta autentificació no és fiable al 100% (sobretot quan tractem amb empremtes parcials) i s'han donat casos on aquesta falta de fiabilitat ha produït errors com condemnes de persones innocents (vegeu [23]).

1.3 Funcionament dels sistemes biomètrics

Els processos biomètrics consten de dues fases, inscripció i verificació. En la primera fase, s'obté una mostra del senyal biomètric. D'aquesta mostra se n'extreu una plantilla que es guardarà per a procediments posteriors. Durant la segona fase, s'obté una nova mostra del mateix senyal biomètric i se n'obté una nova plantilla que es compara amb la primera o amb totes les obtingudes prèviament.

Els sistemes biomètrics actuals tenen dos objectius principals: identificació i autentificació. Durant el primer procés, el senyal biomètric d'un individu és comparat amb un conjunt gran de senyals d'altres individus mentre que durant el segon procés l'únic que es demana és la comparació de dos senyals per afirmar o negar la seva igualtat. El primer procés (també anomenat *one-to-many*) pot ser útil per determinar la procedència d'una empremta dactilar trobada, per exemple, en algun escenari policial. En canvi, el segon (*one-to-one*) pot ser usat en l'accés físic a determinades zones o edificis i també en l'accés virtual per entrar a certs sistemes informàtics on l'usuari presenta la seva identitat i, per confirmar-ho, introdueix el seu senyal biomètric. Un terme mig entre aquests processos és l'anomenat *one-to-few* (entre 2 i 10^4) que és útil, per exemple, en alguna empresa o zona restringida on l'accés només està garantit per a un nombre reduït de persones. Quan un individu vol accedir a la zona en qüestió, presenta el seu senyal biomètric, si aquest coincideix amb algun dels emmagatzemats és acceptat i, en cas contrari, és rebutjat.

1.4 L'ús actual de les claus criptogràfiques

Degut als potents sistemes informàtics actuals, les claus criptogràfiques segures han de tenir una mida molt gran, fet que impossibilita que l'usuari pugui recordar la seva clau. A més a més, quan volem accedir a més d'un lloc diferent, per garantir la seguretat hem de recordar tantes claus com llocs, fet que dificulta encara més la tasca. Normalment, el que es fa avui en dia és xifrar aquesta clau de manera que per poder accedir a ella l'usuari només hagi d'introduir un PIN o contrasenya fàcil de recordar. La clau privada és emmagatzemada en algun emplaçament de difícil accés. Però aquest sistema presenta dos riscs bàsics, el primer es deriva del fet que, els usuaris tenen tendència a usar PIN's senzills, relacionats amb dades personals o fins i tot se'ls apunten en un paper (sobretot quan n'han de recordar un nombre considerable) fet que els fa poc segurs. El segon risc és una conseqüència de la falta de connexió entre el PIN i l'usuari, que fa que l'algorisme que llegeix el PIN no és capaç de saber si la persona que està entrant el PIN és l'usuari legítim o no.

Una alternativa per a la protecció de claus criptogràfiques (o de dades en general) és l'ús de la biometria. Quan l'usuari vol accedir a aquesta clau, proporciona al sistema (mitjançant el sensor, escàner o lector adequat) una mostra del seu senyal biomètric. Si la verificació és correcte, l'usuari és autoritzat i pot accedir a la seva clau criptogràfica amb la qual podrà, per exemple, xifrar o desxifrar les dades desitjades. Aquest sistema aboliria la utilització de PIN's i solucionaria el problema del lligam entre el PIN i l'usuari.

1.5 El tractament de la informació

Els senyals biomètrics s'usen per identificar-nos, per saber qui som, però moltes vegades necessitarem acreditar-nos mantenint l'anonimat. Per tant presenten un problema de privacitat. Però encara pitjor, presenten un problema de seguretat, ja que podem ser víctimes d'una suplantació de la nostra identitat si ens roben el nostre senyal biomètric.

La creació d'una base de dades nacional d'ADN o d'empremtes dactilars és un tema delicat que ha ocasionat molts debats sobre la privacitat i la intimitat en contra de la seguretat nacional. Per una banda, la creació d'aquest tipus de bases de dades (recolzada per algunes entitats governamentals) asseguraria un major control sobre la població, millorant la seguretat i problemes com la immigració il·legal. Per altra banda el dret a la privacitat dels ciutadans es podria veure compromès i s'hauria de preveure una vulneració o una utilització no desitjada d'aquesta.

La possible vulneració de les bases de dades on es guarda la informació biomètrica dels individus és un tema que pot (i ha de) generar desconfiances. Molts països tenen lleis que responsabilitzen a les empreses de garantir la seguretat de les dades personals dels usuaris,

clients i empleats.

El gran inconvenient de l'ús de la biometria és que un cop interceptada, no només no podem regenerar-la sinó que no la podem recuperar. És a dir, hem perdut la nostra identitat. En canvi, si un PIN o clau secreta ha estat robada, només cal generar-ne un de nou.

1.6 Els sistemes actuals

La biometria presenta un gran repte a l'hora de tractar-la en els camps informàtics i de seguretat ja que les mesures biomètriques sempre són inexactes, és a dir, diverses mesures d'una mateixa empremta dactilar no seran mai iguals. Per tant, a l'hora d'usar els sistemes criptogràfics per protegir-les, un simple *hash* no seria suficient.

Per mesurar la precisió i fiabilitat dels sistemes biomètrics s'introdueixen dos conceptes: la *false rejection rate (FRR)* i la *false acceptance rate (FAR)*. El primer concepte es refereix al percentatge d'individus que seran rebutjats pel sistema tot i ser usuaris legítims, situació en la qual l'usuari hauria de tornar a introduir la mostra biomètrica. El segon es refereix al percentatge d'individus no legítims que seran autoritzats pel sistema.

La majoria dels sistemes actuals tenen rangs de FRR d'entre un $0,1\%$ i un 20% i rangs de FAR d'entre un 1% (en aplicacions amb poca seguretat) i un $0,0001\%$ (en aplicacions que requereixen alts nivells de seguretat). Tot i semblar que són nivells de seguretat acceptables, si pensem en una població de 30.000.000 d'habitants, estaríem davant d'un sistema que provocaria (de mitjana) 30 autoritzacions falses.

Per aconseguir alts nivells de seguretat i alhora protegir les dades biomètriques, s'han dissenyat diferents tècniques i algorismes com per exemple la biometria multimodal.

La majoria dels sistemes actuals usen, per xifrar els fitxers, una clau privada emmagatzemada en un servidor anomenat *segur*. Aquesta clau només és accessible quan la verificació biomètrica s'ha realitzat amb èxit i, a partir d'aquest punt, entra en funcionament algun algorisme criptogràfic convencional. La informació biomètrica és emmagatzemada en algun lloc per a futures verificacions i sovint es xifra amb algun sistema convencional. Aquests sistemes són vulnerables ja que depenen d'una resposta binària (sí/no) per part del verificador i per tant, pot resultar senzill fer canviar el bit de resposta d'aquest. Per altra banda la clau privada no està realment vinculada al senyal biomètric. Això fa que sigui menys segur que altres sistemes on s'estableix un vincle entre aquestes dues magnituds de manera suficientment segura.

Capítol 2

Fuzzy extractors

2.1 Introducció i contextualització

Suposem un context on l'Alice vol usar un dels seus senyals biomètrics (empremta dactilar, iris,...) per xifrar un fitxer. En un primer moment podem pensar que, simplement, es podria fer servir el senyal biomètric com a clau privada. Però aquest fet comportaria dos problemes principals:

- El senyal biomètric de l'Alice no és un senyal uniformement aleatori i les propietats de seguretat de la majoria de sistemes criptogràfics només són vàlids quan la clau es tria de manera uniforme i aleatòria.
- Quan l'Alice vulgui escanejar de nou el seu senyal biomètric, aquest serà similar però no idèntic al primer, per tant, usant un sistema criptogràfic tradicional, l'Alice no podria desxifrar el fitxer ja que la clau estaria modificada.

Per solucionar aquests problemes es defineixen els *fuzzy extractors*. Un *fuzzy extractor* extrau una clau R uniformement aleatòria a partir d'un secret ω que no és uniforme i que pot presentar algunes imperfeccions, és a dir, a partir d'un ω' prou proper a l'ω inicial (en l'espai mètric apropiat) hem de ser capaços de recuperar la nostra clau R. Per dur a terme aquesta tasca el *fuzzy extractor* computa una seqüència d'ajuda P.
Els *fuzzy extractors* consten d'un procediment en dues fases:

- Primera fase: L'usuari introdueix el seu secret ω i el *fuzzy extractor* ens retorna la nostra clau privada R i la seqüència d'ajuda P.
- Segona fase: Quan l'usuari vol recuperar la seva clau torna a introduir el seu secret (potser lleugerament modificat) ω' juntament amb la seqüència P i el *fuzzy extractor* ens retornarà la clau R inicial.

El concepte de *fuzzy extractor* es va introduir en [5] però en la primera versió les comunicacions s'havien de fer en un canal autentificat és a dir, no hi havia cap garantia de seguretat si algú modificava P. Posteriorment en [4] s'introdueix la noció de *fuzzy extractor* robust que soluciona el problema de l'autentificació del canal. Aquest tipus de *fuzzy extractor* garanteix la seguretat encara que la P que introduïm en la segona fase estigui modificada però, en aquesta primera aproximació als *fuzzy extractors* robustos, la clau extreta era significativament més curta que en el cas no robust. A més a més l'entropia de les ω havia de ser massa alta (en principi, s'arriba a un límit teòric de $|W|/2$), fet que els senyals biomètrics no es creu que compleixin.

Finalment, [6] introdueix una nova versió de *fuzzy extractors* robusts on aquests problemes sí que es solucionen, és a dir, la clau extreta és prou llarga (de fet la fita trobada és molt propera al cas no robust de [5]), i a més a més no hi ha cap restricció per a l'entropia de les ω, trencant doncs el límit teòric introduït en [4]. Aquesta millora s'aconsegueix introduint els anomenats paràmetres públics del sistema generats aleatòriament. Aquests paràmetres es trien una vegada el sistema ha estat dissenyat i s'implementen dins d'aquest.

Aquesta definició donada a [3] i [6], està feta en el model CRS (*common random string*) que pressuposa l'existència d'una font segura d'aleatorietat per a la generació dels paràmetres del sistema.

De fet avui en dia els usuaris d'un sistema no el creen ells mateixos sinó que l'adquireixen i, per tant, estan assumint que la implementació està feta correctament (de fet cap sistema criptogràfic seria vàlid si el sistema estigués dissenyat maliciosament). La creença de l'existència d'una font segura d'aleatorietat i d'uns paràmetres públics que no es poden modificar no ens suposa doncs un increment massa gran de confiança.

No obstant, recentment s'ha presentat una nova versió de *fuzzy extractor* robust en [7] que no utilitza paràmetres públics del sistema i que també millora considerablement els resultats de llargada de la clau obtinguts en [4].

2.2 Definicions preliminars

Considerem X i Y variables *hola* aleatòries. Indicarem $x \leftarrow X$ quan x es triï segons la distribució X i $\Pr[X = x]$ voldrà dir la probabilitat assignada a x segons X.

Primer de tot definirem l'entropia mínima d'una variable aleatòria X com:

$$\mathbf{H}_\infty(X) = \min_x(-\log\ \Pr[X = x]) = -\log(\max_x\ \Pr[X = x])$$

Definirem també l'entropia mínima condicional mitjana d'X donada Y com:

$$\tilde{\mathbf{H}}_\infty(X|Y) = -\log(\mathbf{E}_{y\leftarrow Y}(\max_x\ (\Pr[X = x \mid Y = y])) = -\log(\mathbf{E}_{y\leftarrow Y}(2^{-\mathbf{H}_\infty(X|Y=y)}))$$

En aquesta definició l'esperança es calcula sobre tots els y per als quals $\Pr_{Y=y} \neq 0$.
També exposarem una part del Lema 3.1 de [5]:

Lema 2.1. *Si Y té 2^λ elements llavors*

$$\tilde{\mathbf{H}}_\infty(X|Y) \geq \mathbf{H}_\infty((X, Y)) - \lambda$$

Demostració.

$$\begin{aligned}
\tilde{\mathbf{H}}_\infty(X|Y) &= -\log \mathbf{E}_{y\leftarrow Y}(\max_x\ \Pr[X = x \mid Y = y]) = \\
&= -\log \sum_y (\max_x\ \Pr[X = x \mid Y = y]\Pr[Y = y]) = \\
&= -\log \sum_y (\max_x\ \Pr[X = x\ \wedge\ Y = y]) \geq \\
&\geq -\log \sum_y 2^{-\mathbf{H}_\infty((A,B))} = -\log 2^\lambda 2^{-\mathbf{H}_\infty((A,B))} = \mathbf{H}_\infty((A, B)) - \lambda
\end{aligned}$$

□

Siguin ara X_1 i X_2 dues distribucions de probabilitat sobre S. La seva distància estadística es defineix com:

$$\mathbf{SD}(X_1, X_2) = \frac{1}{2}\sum_{s\in S} |\Pr[X_1 = s] - \Pr[X_2 = s]|$$

Si dues variables aleatòries estan a distància com a molt ε diem que són ε-properes i escrivim $X_1 \approx_\varepsilon X_2$.

Finalment recordarem les funcions de *hash*: una funció $H : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^l$ és δ-quasi universal si per a qualssevol x, x' diferents, tenim $\Pr_{i\leftarrow U_d}[H(x, i) = H(x', i)] \leq \delta$. Les famílies per a les quals $\delta = 2^{-l}$ es diuen universals.

2.3 Parts d'un fuzzy extractor

En primer lloc definirem els *randomness extractors* que, tal i com el seu nom indica, ens permetran obtenir dades uniformement aleatòries a partir d'altres que no ho són (o tenen perquè ser-ho).

Una funció probabilística eficient $\mathbf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^l$ és un (m, l, ε)-*extractor* fort si per a totes les distribucions X sobre $\{0,1\}^n$ amb $\mathbf{H}_\infty(X) \geq m$ tenim que:

$$\mathbf{SD}((\mathbf{Ext}(X; U_d), U_d), U_{l+d}) \leq \varepsilon$$

Els *randomness extractors* es poden construir a partir de funcions de *hash* universals tal i com expressa el Lema 3.1 de [6].

Lema 2.2. *Si A i B són dues variables aleatòries tals que $A \in \{0,1\}^n$ i $\tilde{\mathbf{H}}_\infty(A|B) \geq m$ i $H : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^l$ és una funció de hash universal, llavors:*

$$\mathbf{SD}((B, U_d, H(A, U_d)), (B, U_d, U_l)) \leq \varepsilon$$

sempre que $l \leq m - 2\log(\frac{1}{\varepsilon}) + 2$.

També definirem els *secure sketches* que ens permetran recuperar qualsevol seqüència $\omega \in \mathcal{M}$, on $\mathcal{M}$ és l'espai mètric on treballem, a partir d'un $\omega' \in \mathcal{M}$ prou proper a ω (segons la distància que considerem en $\mathcal{M}$) difonent la mínima informació d'ω possible.
Així doncs, direm que un (m, m', t)-*secure sketch* per a un espai mètric qualsevol està format per un parell de procediments eficients (**SS**, **Rec**) que compliran les següents propietats:

- El procediment **SS** rebrà com a input $\omega \in \mathcal{M}$ i retornarà una seqüència de bits $s \in \{0,1\}^*$.
- El procediment **Rec** agafarà com a input un element $\omega' \in \mathcal{M}$ i $s \in \{0,1\}^*$. Per a la correctesa establirem que si $\mathbf{dis}(\omega, \omega') \leq t$ on **dis** és la distància considerada en l'espai mètric $\mathcal{M}$ llavors $\mathbf{Rec}(\omega', \mathbf{SS}(\omega)) = \omega$.
- Seguretat: Per a qualsevol distribució W sobre $\mathcal{M}$ tenim que si $\mathbf{H}_\infty(W) \geq m$ llavors $\tilde{\mathbf{H}}_\infty(W, \mathbf{SS}(W)) \geq m'$.

El valor de $m - m'$ s'anomena entropia perduda del *secure sketch*.
Si prenem com a espai mètric el de Hamming, és a dir, $\mathcal{M} = \{0,1\}^n$ es pot construir un *secure sketch* a partir d'una construcció amb síndromes i codis correctors d'errors. Aquesta construcció està feta a [5].

El cas dels *fuzzy extractors* robustos, degut a que la robustesa està molt lligada a l'autentificació, requerirà l'ús de codis d'autentificació de missatges (MAC). Aquests codis, donat un input (x, k) (on x és l'entrada pública de l'usuari i k és una clau privada) assignaran una etiqueta σ per tal que un adversari no pugui manipular $(x, \sigma) \to (x', \sigma')$ de manera que aquest nou input manipulat sigui vàlid per a alguna clau k'.

Així doncs definim que una funció **MAC**: $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\nu$ és un MAC δ-segur amb seguretat forta si per a qualsevol $x \neq x'$ i qualssevol σ, σ'

$$\Pr[\mathbf{MAC}(x'; k) = \sigma' | \mathbf{MAC}(x; k) = \sigma] \leq \delta$$

on la probabilitat es pren sobre $k \leftarrow U_d$.

En particular, per a la construcció dels *fuzzy extractors* usarem MACs amb seguretat contra manipulacions de la clau, aquests s'anomenen KMS-MACs i com el seu nom indica afegeixen als MAC seguretat contra possibles manipulacions de la clau per part de l'adversari. És a dir, donem a l'adversari la capacitat de, sense conèixer la clau, fer modificacions que puguin afectar-la $k \to k + \Delta$ (fins i tot coneixent aquesta diferència Δ).

Direm que un KMS-MAC és δ-segurs amb seguretat forta si per a qualsevol $x \neq x'$ i qualssevol σ, σ' i Δ,

$$\Pr[\mathbf{MAC}(x'; k + \Delta) = \sigma' | \mathbf{MAC}(x; k) = \sigma] \leq \delta$$

on la probabilitat es pren sobre $k \leftarrow U_d$.
La construcció dels MAC i dels KMS-MAC està molt relacionada amb l'estudi dels AMD-*codes* (codis de detecció contra manipulacions algebraiques) que veurem més endavant.

2.4 Construcció teòrica d'un fuzzy extractor

Com ja hem explicat, podem construir *fuzzy extractors* robustos i no robustos. La diferència principal és que en el cas no robust no hi ha garanties de seguretat si modifiquem la seqüència d'ajuda P. De *fuzzy extractors* robustos n'hi ha que usen els paràmetres del sistema i d'altres no els usen. Com explicarem més endavant els *fuzzy extractors* no robustos no necessiten usar paràmetres del sistema.

Primer de tot exposarem una versió de *fuzzy extractor* no robust sense usar paràmetres del sistema, després veurem el cas robust amb les modificacions pertinents per permetre la incorporació d'aquests paràmetres.

Un (m, l, t, ε)- *fuzzy extractor* per a un espai mètric qualsevol $\mathcal{M}$ està format per un parell de procediments eficients amb certes components aleatòries. A aquests procediments els anomenarem (**Gen**, **Rep**) i compliran les següents propietats:

- El procediment **Gen** rebrà com a input $\omega \in \mathcal{M}$ i extreurà una clau $R \in \{0,1\}^l$ i una seqüència d'ajuda $P \in \{0,1\}^*$.

- El procediment **Rep** rebrà com a inputs $\omega' \in \mathcal{M}$ i una seqüència $P \in \{0,1\}^*$. Per a la correctesa establirem que si $\mathbf{dis}(\omega, \omega') \leq t$ on **dis** és la distància considerada en l'espai mètric $\mathcal{M}$ i $\mathbf{Gen}(\omega) = (R, P)$ llavors $\mathbf{Rep}(\omega', P) = R$.

- Seguretat: Per a qualsevol distribució W sobre $\mathcal{M}$ tenim que si $\mathbf{H}_\infty(W) \geq m$ i $\mathbf{Gen}(W) = (R, P)$ llavors $\mathbf{SD}((R, P), (U_l, P)) \leq \varepsilon$.

Composant un (m, m', t)-*secure sketch* amb un (m', l, ε)-*extractor* fort obtenim un (m, l, t, ε)-*fuzzy extractor* amb $P = (\mathbf{SS}(\omega), i)$ i $R = \mathbf{Ext}(\omega, i)$.

Els *fuzzy extractors* permeten fer pública la seqüència d'ajuda P però, en aquesta versió en particular no hi ha cap garantia de seguretat si un adversari actiu modifica P. Per tant, ara voldrem que per a qualsevol valor $\tilde{P} \neq P$ produït per l'adversari, $\mathbf{Rep}(\omega', \tilde{P}) = \perp$ amb probabilitat d'error el més baixa possible. D'aquesta manera detectarem quan l'adversari ha intentat modificar P i tindrem la robustesa que volíem.

En [4] es fa una distinció dels *fuzzy extractors* robustos depenent de la informació que té l'adversari quan vol modificar P. Així doncs, es distingeix entre robustesa amb *pre-application* o amb *post-application* depenent de si l'adversari té accés a R o no, respectivament, a l'hora de modificar P.

Formalment, considerem (W, W') dues variables aleatòries sobre un espai mètric $\mathcal{M}$, direm que formen un (t, m)-parell correlacionat si $\mathbf{dis}(W, W') \leq t$ i $\mathbf{H}_\infty(W) \geq m$.
Diem que un $(m, l, t, \varepsilon, \delta)$- *fuzzy extractor* té robustesa amb *post-application* si per a tots els (t, m)-parells correlacionats (W, W') i per a tots els adversaris $\mathcal{A}$:

$$\Pr\left[\begin{array}{c|l} \mathbf{Rep}(\tilde{P}, \omega') \neq \perp & (\omega, \omega') \leftarrow (W, W') \\ \tilde{P} \neq P & (P, R) \leftarrow \mathbf{Gen}(\omega) \\ & \tilde{P} \leftarrow \mathcal{A}(P, R) \end{array}\right] \leq \delta$$

El cas de robustesa amb *pre-application* es defineix anàlogament però denegant l'accés de R a l'adversari $\mathcal{A}$. Així doncs, definim ara la implementació usant paràmetres del sistema d'un *fuzzy extractor* robust.

Un $(m, l, t, \varepsilon, \delta)$- *fuzzy extractor* robust en un espai mètric $\mathcal{M}$ consisteix en tres procediments: **Init**, **Gen** i **Rep**, amb certes components aleatòries. Aquests procediments compleixen les següents propietats:

- El procediment **Init** no té cap input i treu com a output els paràmetres del sistema Params $\in \{0,1\}^*$.

- El procediment **Gen** té com a inputs $\omega \in \mathcal{M}$ i Params $\in \{0,1\}^*$ i com a outputs les seqüències $R \in \{0,1\}^l$ i $P \in \{0,1\}^*$
- El procediment **Rep** té com a inputs $\omega' \in \mathcal{M}$, $P \in \{0,1\}^*$ i Params $\in \{0,1\}^*$. Per a la correctesa voldrem que si $\mathbf{dis}(\omega, \omega') \leq t$ i $\mathbf{Gen}(\omega, \text{Params}) = (P, R)$ llavors $\mathbf{Rep}(\omega', P, \text{Params}) = R$
- La seguretat requerirà que per a qualsevol distribució W sobre $\mathcal{M}$ amb $\mathbf{H}_\infty(W) \geq m$, si **Init**()=Params i $\mathbf{Gen}(W,$ Params)$=(R, P)$ llavors s'haurà de complir que $\mathbf{SD}((R, P, \text{Params}), (U_l, P, \text{Params})) \leq \varepsilon$
- La robustesa amb *post-application* imposarà que per a tots els (t, m)-parells correlacionats (W, W') i tots els adversaris $\mathcal{A}$:

$$\Pr\left[\begin{array}{c|c} \mathbf{Rep}(\tilde{P}, \omega') \neq \perp & \text{Param} \leftarrow \mathbf{Init}() \\ \tilde{P} \neq P & (\omega, \omega') \leftarrow (W, W') \\ & (P, R) \leftarrow \mathbf{Gen}(\omega, \text{Params}) \\ & \tilde{P} \leftarrow \mathcal{A}(\text{Params}, P, R) \end{array}\right] \leq \delta$$

  El cas de la robustesa amb *pre-application* es definirà de manera anàloga però sense donar R a l'adversari.

Aquest construcció teòrica (que inclou els paràmetres del sistema) també es pot aprofitar en el cas no robust ja que només cal introduir la generació d'aquests com a part del procediment **Gen** i incloure'ls dins la seqüència P.

2.5 Construcció pràctica d'un fuzzy extractor

En aquesta secció exposarem els procediments **Init**, **Gen** i **Rep** en detall per construir un *fuzzy extractor* robust fent servir un *secure sketch* (**SS**, **Rec**), un *randomness extractor* (**Ext**) i un **MAC**. Els passos a seguir són els que s'exposen a continuació (també esquematitzats en les Figures 2.1 i 2.2):

- **Init()** extreu una llavor aleatòria i que representa el conjunt de paràmetres del sistema.
- **Gen** (ω, i)
  1. R=**Ext**(ω, i) i separem R de la següent manera $R = (R_{mac}, R_{out})$.
  2. $s = \mathbf{SS}(\omega)$, $\sigma = \mathbf{MAC}(s, R_{mac})$ i posem $P = (s, \sigma)$
  3. Output $= (P, R_{out})$

- **Rep** $(\omega', \tilde{P})$

  1. Separem $\tilde{P} = (\tilde{s}, \tilde{\sigma})$
  2. $\tilde{\omega} =$**Rec**$(\omega', \tilde{s})$. Si **dis**$(\tilde{\omega}, \omega') > t$ llavors responem $\perp$ i parem. En cas contrari seguim endavant.
  3. $\tilde{R} = (\tilde{\omega}, i)$ i tornem a separar $\tilde{R} = (\tilde{R}_{mac}, \tilde{R}_{out})$
  4. Si $\tilde{\sigma} =$ **MAC**$(\tilde{s}, \tilde{R}_{mac})$ responem $\tilde{R}_{out}$. En cas contrari responem $\perp$.

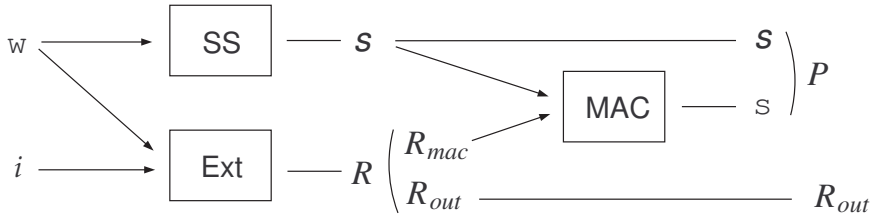


Figura 2.1: Procediment **Gen** (ω, i)

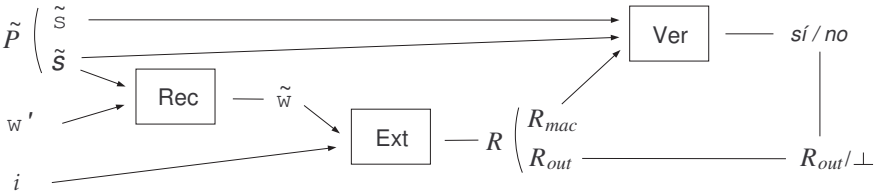


Figura 2.2: Procediment **Rep** $(\tilde{\omega}, \tilde{P})$

Cal notar que aquesta construcció té un problema de seguretat. Si un adversari modifica $s \to \tilde{s}$ això afectarà a la incorrecta recuperació d'ω, és a dir, el procediment **Rec** extreurà $\tilde{\omega}$ i llavors **Ext** ens respondrà $\tilde{R} = (\tilde{R}_{mac}, \tilde{R}_{out})$. Per tant, la modificació de s influencia en la modificació de la clau del **MAC**, que només garantia seguretat si modificàvem la primera entrada del **MAC** (en aquest cas s).

A més a més, es pot demostrar (està fet a [6]) que, fent servir procediments lineals (fet que és el més habitual), un adversari que conegui $\Delta = \omega' - \omega$ i modifiqui $P = (s, \sigma)$, pot controlar amb una funció determinista la diferència $\Delta_R = \tilde{R} - R$. Per tant, l'adversari no només influencia en la modificació de la clau del **MAC** sinó que a més coneix exactament el valor d'aquesta modificació.

Aquests fets ens fan perdre la seguretat el **MAC** i, per solucionar-ho farem servir **KMS-MAC**s, que sí que garanteixen la seguretat en cas que es modifiqui la clau.

2.6 Fites i llargades de la clau extreta

El problema principal dels *fuzzy extractors* és poder extreure una clau R uniformement aleatòria i suficientment llarga a partir d'unes dades que no tenen perquè ser-ho i que poden presentar un valor d'entropia qualsevol.

En la primera aparició dels *fuzzy extractors* en [5] es presenta un (m, l, t, ε)-*fuzzy extractor* no robust construït a partir d'un (m, m', t)-*secure sketch* i d'un (m', l, ε)-*randomness extractor*. Aquest *fuzzy extractor* aconseguia extreure una clau R de llargada l on:

$$l \leq m' - 2\log\left(\frac{1}{\varepsilon}\right) + O(1)$$

Posteriorment, amb l'aparició dels *fuzzy extractors* robustos en [4], la llargada de la clau secreta disminueix considerablement. En aquest cas s'obté un $(m, l, t, \varepsilon, \delta)$-*fuzzy extractor* robust on les fites per a la clau són:

$$l \leq 2m - n - u - 2t\log\left(\frac{en}{t}\right) - 2\log\left(\frac{n}{\varepsilon^2\delta}\right) - O(1)$$

si tractem amb robustesa amb *pre-application* o

$$l \leq \frac{1}{3}\left(2m - n - u - 2t\log\left(\frac{en}{t}\right) - 2\log\left(\frac{n}{\varepsilon^2\delta}\right)\right) - O(1)$$

en el cas de robustesa amb *post-application*.
En aquestes fites u i n són les llargades en bits de $\mathbf{SS}(\omega)$ i de $\mathcal{M}$ respectivament.

Finalment, en [6] (amb la incorporació de paràmetres públics del sistema) les fites aconseguides per a un $(m, l, t, \varepsilon, \delta)$-*fuzzy extractor* robust són:

$$l \leq m' - 2\log\left(\frac{u}{\varepsilon\delta}\right) - 2$$

amb la *pre-application robustness* on $m' = \tilde{\mathbf{H}}_\infty(W|\mathbf{SS}(W))$ o

$$l \leq \hat{m} - 2\log\left(\frac{u}{\varepsilon\delta}\right) - 2$$

amb la *post-application robustness* on $\hat{m} = \tilde{\mathbf{H}}_\infty(W|\mathbf{SS}(W), W - W')$.

Aquestes fites són molt més properes al cas no robust anterior que no pas al cas robust de [4]. Per tant, podem concloure que la inclusió d'aquests paràmetres (que com ja hem dit és un fet bastant natural) permet que el cost d'afegir robustesa al nostre sistema sigui molt petit.

2.7 Avantatges i inconvenients dels fuzzy extractors. Ús d'altres sistemes mètrics

Un dels principals inconvenients dels *fuzzy extractors* és la manca d'una fase per al tractament de la imatge. És a dir, si volguéssim implementar un *fuzzy extractor* en un sistema biomètric hauríem d'idear un sistema per tractar les imatges que ens arribessin de l'escàner.

En aquest sentit el mètode de la *Biometric Encryption* que veurem en el capítol següent està més desenvolupat degut al fet que per al tractament de la imatge s'ha dissenyat un sistema que fa servir transformades de Fourier. Aquest sistema usa tota la imatge, en aquest cas d'empremtes dactilars, en comptes d'usar algun mètode específic per a aquest senyal biomètric com ara les minúcies.

Mentre que el sistema *Biometric Encryption* comença a actuar en el moment que l'usuari introdueix el seu senyal biomètric a l'escàner , els *fuzzy extractors* no comencen fins que aquest senyal ha estat transformat en una seqüència de bits ω. Però aquesta transformació es pot dur a terme de moltes maneres (llegir la imatge directament, fer servir minúcies, transformades de Fourier,...) i haurem d'adaptar la mètrica d'aquestes transformacions amb la que farem servir al nostre sistema.

Per exemple, situem-nos en un context on llegim una empremta dactilar. Suposem que llegim tota la imatge i transcrivim un bit 0 si el píxel és blanc i un bit 1 en cas contrari (podem suposar que la imatge està en blanc i negre). Aquest tractament de la imatge no seria coherent per exemple, amb la distància de Hamming ja que, si apliquem una petita translació a la imatge, la seva transformació a una seqüència de bits podria distar molt de la seqüència extreta inicialment. Per tant hauríem d'establir un altre sistema mètric que fos coherent amb els moviments i deformacions que pot presentar l'escanejat del nostre senyal biomètric o canviar el tipus de tractament de la imatge que realitzem.

Evidentment, aquestes consideracions variaran en funció del tipus de senyal biomètric que usem i també en funció del tipus d'escanejat que fem servir.

En [5] es treballa amb altres distàncies a més a més de la distància de Hamming i fins i tot es construeix un *fuzzy extractors* usant la distància d'edició. Però el cert és que en tots els altres articles més recents s'han deixat de banda aquestes altres distàncies per centrar-se bàsicament en la de Hamming (que és la que aporta les construccions per síndrome usant codis correctors d'errors). Aquesta centralització podria portar problemes a l'hora d'implementar aquest sistema a la pràctica ja que, com ja hem dit, no està clar quin sistema de tractament de la imatge s'ha de fer servir (si existeix) per tal que la distància de Hamming sigui coherent amb aquest.

Un altre inconvenient dels *fuzzy extractors* el trobem en la fase per desxifrar ja que,

durant aquesta fase es recupera l'ω i, per tant recuperem l'input del *fuzzy extractor*. Si aquest input era el nostre senyal biomètric o alguna seqüència de bits a partir de la qual podem recuperar-lo o extreure'n informació, estarem posant en perill la confidencialitat del nostre senyal biomètric en cas que l'adversari tingui accés a aquesta informació.

Capítol 3

Biometric encryption

3.1 Introducció i contextualització

Una altra manera de tractar els senyals biomètrics ha estat desenvolupada per Mytec Technologies Inc. (vegeu [2] i [13]). La idea és la mateixa que la usada en els *fuzzy extractors* és a dir, usar un senyal biomètric per poder xifrar i desxifrar missatges de forma segura i sense haver de recordar cap PIN o contrasenya.

Però aquest mètode difereix bastant, pel que fa a procediments, dels *fuzzy extractors*. En primer lloc aquest mètode usa una clau criptogràfica que en un principi no està relacionada amb el senyal biomètric. En canvi, en els *fuzzy extractors* la clau és extreta a partir del senyal biomètric amb un *randomness extractor*. Ara bé, en la *Biometric encryption*, un cop s'ha establert la connexió entre la clau i el senyal (en un principi independents l'una de l'altre) s'uneixen de manera que es pugui recuperar la clau a partir d'un nou escàner del senyal que, igual que en el cas dels *fuzzy extractors*, es considera semblant però no idèntic al primer. Aquest procediment també es fa de manera que un adversari qualsevol no pugui recuperar el senyal biomètric a partir de les dades emmagatzemades (o ho pugui fer amb probabilitat molt petita). En aquest sistema les dades emmagatzemades reben el nom de *Bioscrypt*.

Una altra diferència bàsica comentada anteriorment, és que el mètode de la *Biometric encryption* està bastant més desenvolupat i incorpora un sistema de tractament de la imatge basat en transformades de Fourier. Aquest mètode aprofita tota la imatge de l'empremta dactilar (o senyal biomètric tractat) en comptes d'usar algun tractament específic, com per exemple les minúcies en el cas de les empremtes.

El sistema de la *Biometric encryption* es basa, com tots els sistemes biomètrics, en dues fases: la d'inscripció i la de verificació.

Per a la construcció d'aquest sistema, necessitarem conèixer prèviament dos procediments auxiliars. El primer és el filtre, que serà creat durant la fase d'inscripció i el segon

serà el *link algorithm* que es farà servir en la fase d'inscripció per unir la clau criptogràfica amb el senyal biomètric i durant la fase de verificació per obtenir la clau criptogràfica a partir de les dades emmagatzemades i la nova lectura del senyal biomètric.

3.2 Construcció del filtre

Per a la construcció del filtre usarem unes imatges de prova del senyal biomètric de l'usuari obtingudes durant la fase d'inscripció que denotarem amb els subíndexs 0. Posteriorment, durant la fase de verificació farem servir els nous escanejats del mateix senyal biomètric per recuperar la clau criptogràfica inicial. Aquests nous valors obtinguts els denotarem amb els subíndexs 1. A més a més, farem servir diverses funcions juntament amb les seves corresponents transformades de Fourier les quals indicarem amb la mateixa lletra que la funció inicial però amb majúscules.

Durant la fase d'inscripció demanarem a l'usuari que introdueixi el seu senyal biomètric $T \geq 1$ vegades. Aquestes T entrades les podem denotar com $\{f_0^1(x), f_0^2(x), \ldots, f_0^T(x)\}$, on la variable x s'està considerant bidimensional és a dir, aquestes funcions es poden pensar com la descripció d'una imatge on, a cada posició x, tenim assignat un valor que podria correspondre per exemple, a una escala de grisos. Aquestes funcions ens serviran per crear el filtre que usarem posteriorment (durant la fase de verificació) per autentificar aquest usuari a través d'unes altres lectures del seu senyal biomètric $f_1^i(x)$.

Considerem $H(u)$ el filtre que volem trobar i anomenem $F_0^i(u)$ les transformades de Fourier de $f_0^i(x)$ $(i = 1 \ldots T)$. Així doncs, $C_0^i(u) := F_0^i(u)H(u)$ on $C_0^i(u)$ són les transformades de Fourier de les funcions que anomenarem patró $c_0^i(x)$, les quals seran obtingudes pel nostre algorisme a partir de les $f_0^i(x)$.

Aquestes funcions $c_0^i(x)$ voldrem que minimitzin la distància a una funció desitjada $r(x)$ que triarem a posteriori per garantir la seguretat del sistema. En altres paraules, voldrem minimitzar la funció següent:

$$E_s = \frac{1}{T} \sum_{i=1}^{T} \int |c_0^i(x) - r(x)|^2 dx$$

Aquest terme ens indica quin grau d'exigència demanem al filtre és a dir, controla la semblança entre la resposta de l'algorisme i la funció $r(x)$ (ja que nosaltres voldríem $c_0^i(x) \approx r(x)$ per a qualsevol i).

A més a més, també voldrem minimitzar l'error degut a la distorsió de les imatges d'entrada. Aquest terme s'expressara de la manera següent:

$$E_n = \int |H(u)|^2 P(u) du$$

on $P(u)$ és la densitat d'espectre de les diferències entre les imatges $f_0^i(x)$ inicials, és a dir:

$$P(u) = \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} |FT(f_0^i(x) - f_0^j(x))|^2$$

Considerant els dos errors a minimitzar tenim que l'error total que volem minimitzar és:

$$E_t = \alpha E_n + \sqrt{1-\alpha^2} E_s$$

on α és un coeficient entre 0 i 1 que ens permetrà optimitzar el funcionament del filtre. Substituint les equacions dels errors en aquesta última i minimitzant l'error total en funció d'$H(u)$ arribem a la següent expressió del filtre:

$$H(u) = \frac{A_0^*(u)R(u)}{\alpha P(u) + \sqrt{1-\alpha^2} D_0(u)}$$

on * denota el conjugat complex i on

$$A_0(u) = \frac{1}{T} \sum_{i=1}^{T} F_0^i(u)$$

$$D_0(u) = \frac{1}{T} \sum_{i=1}^{T} |F_0^i(u)|^2$$

és a dir, $A_0(u)$ i $D_0(u)$ es poden interpretar com mitjanes relacionades amb les $F_0^i(u)$. A més a més, com ja havíem comentat el filtre també incorpora un coeficient $\alpha \in [0, 1]$ que ens indicarà el grau de tolerància que li permetem. Com més gran sigui aquest valor, més permissiu serà amb els errors de distorsió de l'input però podrà extreure un senyal filtrat que s'allunyarà més de les funcions patró que hem introduït al principi.

La funció $R(u)$ és la transformada de Fourier de la funció $r(x)$ esmentada anteriorment. Per maximitzar la seguretat del filtre, la funció $R(u)$ es prendrà uniformement aleatòria i de mòdul 1 (recordem que és una funció complexa). És a dir:

$$R(u) = e^{i\phi_R(u)}$$

D'aquest filtre $H(u)$ que es dissenyarà amb les primeres T imatges del senyal, només se'n guardarà una part al *Bioscrypt* anomenada H_{stored}, per a més seguretat.

$$H_{stored} = e^{-i\phi_{A_0}(u)} e^{i\phi_R(u)}$$

Fixem-nos que el fet que $R(u)$ sigui aleatòria fa que un adversari no pugui recuperar cap informació sobre $A_0(u)$ veient H_{stored}.

Aquesta informació emmagatzemada no ens permetrà recuperar el filtre original però, juntament amb una nova imatge $f_1(x)$ del mateix senyal biomètric, serà capaç de filtrar aquest nou valor i obtenir la funció $c_1(x)$ corresponent. A més a més tampoc no ens permetrà recuperar ni el valor filtrat inicial $c_0(x)$ ni els senyals biomètrics inicials $f_0^i(x)$. Veiem amb més detall com es duu a terme aquesta operació.

Primerament, tenint en compte que tant $P(u)$ com $D_0(u)$ són funcions reals i positives el filtre $H(u)$ queda separat de la següent manera:

$$H(u) = |H(u)|e^{i\phi_H(u)} = \frac{|A_0(u)|}{\alpha P(u) + \sqrt{1-\alpha^2}D_0(u)} e^{-i\phi_{A_0}(u)} e^{i\phi_R(u)}$$

Ara suposem que hem introduït les imatges $f_0^i(x)$ a l'algorisme i que hem calculat el filtre $H(u)$. A continuació trobem la funció $c_0(x) = FT^{-1}\{A_0(u)H(u)\}$ que representa el valor mitjà de les funcions $c_0^i(x) = FT^{-1}\{F_0^i(u)H(u)\}$.

$$\begin{aligned} c_0(x) &= FT^{-1}\left\{ A_0(u) \frac{|A_0(u)|}{\alpha P(u) + \sqrt{1-\alpha^2}D_0(u)} e^{-i\phi_{A_0}(u)} e^{i\phi_R(u)} \right\} = \\ &= FT^{-1}\{A_0(u) \cdot |H_0(u)| \cdot H_{stored}(u)\} \end{aligned}$$

Un cop calculat $c_0(x)$ només emmagatzemarem H_{stored}. Durant la fase de verificació obtindrem nous escanejats del nostre senyal biomètric $f_1^j(x)$ ($A_1(u)$ si treballem amb mitjana i amb transformades de Fourier). Amb aquest nou valor i aproximant $|A_0(u)|$ per $|A_1(u)|$ i $D_0(u)$ per $D_1(u)$ podem fer els següents càlculs per aconseguir la nova imatge filtrada $c_1(x)$:

$$\begin{aligned} c_1(x) &= FT^{-1}\left\{ A_1(u) \frac{|A_1(u)|}{\alpha P(u) + \sqrt{1-\alpha^2}D_1(u)} e^{-i\phi_{A_0}(u)} e^{i\phi_R(u)} \right\} = \\ &= FT^{-1}\{A_1(u) \cdot |H_1(u)| \cdot H_{stored}(u)\} \end{aligned}$$

Per tant amb $H_{stored}(u)$ que havíem emmagatzemat i $A_1(u)$ i $|H_1(u)|$ que podem calcular amb els nous escanejats $f_1^j(x)$, obtenim la imatge filtrada que volíem $c_1(x)$.

3.3 El link algorithm

Aquesta part del sistema de la *Biometric encryption* permet lligar i recuperar la clau criptogràfica k_0 amb l'ajuda del senyal filtrat $c_0(x)$. La clau criptogràfica k_0 és la que voldrem fer servir per xifrar i desxifrar els nostres missatges i/o fitxers.

Durant la fase d'inscripció lligarem la nostra clau k_0 amb $c_0(x)$ que, com ja hem vist, és una mitjana dels valors filtrats obtinguts a partir de les imatges de prova i crearem una taula $\mathbf{M}$ que emmagatzemarem al *Bioscrypt*. Posteriorment, per recuperar la nostra clau criptogràfica durant la fase de verificació, haurem d'utilitzar la taula $\mathbf{M}$ i el senyal filtrat $c_1(x)$ obtingut a partir del nou escanejat del senyal biomètric $f_1(x)$ i del valor emmagatzemat del filtre H_{stored}. Aquesta taula que farem servir tindrà tantes columnes com bits tingui k_0 i L files on L serà un valor que haurà de complir uns certs requisits que exposarem més endavant un cop s'hagi vist la funció de la taula $\mathbf{M}$ dins del procediment.

Notem que $c_0(x)$ i $c_1(x)$ no seran iguals ja que els escanejats del senyal biomètric seran semblants però no idèntics. Per tant, la taula que farem servir haurà de contenir certa redundància. Tanmateix, un adversari que vegi només aquesta taula $\mathbf{M}$ no podrà recuperar ni la clau k_0 ni el valor filtrat c_0. Ara bé, un usuari que tingui $c_1(x)$ i la taula podrà recuperar k_1 que en un principi hauria de coincidir amb la clau inicial k_0.

El procediment que fa servir aquest algorisme es basa fonamentalment en un senzill codi de repetició. En primer lloc s'agafa una part de la imatge filtrada $c_0(x)$ i es converteix en una taula de valors binaris $\mathbf{B_0}$ (procediment que explicarem amb detall més endavant).

Seguidament, s'agafa el primer bit b de la clau criptogràfica k_0. Es trien a l'atzar L coordenades diferents $(x1_i, y1_i)$ de $\mathbf{B_0}$ que continguin aquest bit b i amb aquestes coordenades omplim la primera columna de la taula $\mathbf{M}$. Així doncs $\mathbf{M}(i,1) = ((x1_i, y1_i))$ per a qualsevol $i = 1 \dots L$. Després agafarem el segon bit b' de k_0 i amb el mateix procediment omplim la segona fila de la taula $\mathbf{M}$ amb els valors $(x2_i, y2_i)$. Un cop completada aquesta operació amb tots els bits de k_0 tindrem plena la taula $\mathbf{M}$ que emmagatzemarem com a part del *Bioscrypt*. Finalment, descartarem tant la clau k_0 com el filtrat $c_0(x)$ per garantir la seguretat del sistema.

Per recuperar la clau criptogràfica k_1 (en principi igual a k_0) a partir de la taula $\mathbf{M}$ i del nou filtrat $c_1(x)$ seguirem el següent procediment. Primerament, convertirem $c_1(x)$ en una taula de valors binaris $\mathbf{B_1}$ de la mateixa manera que ho havíem fet amb $\mathbf{B_0}$. Seguidament ens fixarem en les coordenades de la primera columna de la taula $\mathbf{M}$, $(x1_i, y1_i)$, i les situarem en $\mathbf{B_1}$. Anotarem quants d'aquests punts corresponent a un bit 0, és a dir $|\{i \ / \ \mathbf{B_1}(x1_i, y1_i) = 0\}|$ i quants corresponen a un bit 1, és a dir $|\{j \ / \ \mathbf{B_1}(x1_j, y1_j) = 1\}|$. Finalment, el primer bit de k_1 serà el bit que aparegui més vegades. Aquesta operació la repetirem fins a completar tots els bits de k_1 i d'aquesta manera recuperarem la nostra clau criptogràfica.

Esquema per a la fase d'inscripció (amb $L = 3$):

$$k_0 = (*, 0, 1, \cdots, 0)$$
$$\downarrow$$
$$\mathbf{B}_0 = \begin{pmatrix} * & 1 & \cdots & 0 \\ 0 & * & \ddots & * \\ \vdots & & & \end{pmatrix} \rightarrow \mathbf{M} = \begin{pmatrix} (a_1, b_1) & \cdots \\ (a_2, b_2) & \\ (a_3, b_3) & \end{pmatrix}$$

on (a_i, b_i) corresponen a les coordenades dels bits $*$ de $\mathbf{B}_0$

Esquema per a la fase de verificació (amb $L = 3$):

$$\mathbf{M} = \begin{pmatrix} (a_1, b_1) & \cdots \\ (a_2, b_2) & \\ (a_3, b_3) & \end{pmatrix}$$
$$\downarrow$$
$$\mathbf{B}_1 = \begin{pmatrix} * & 1 & \cdots & 0 \\ 1 & * & \ddots & * \\ \vdots & & & \end{pmatrix} \rightarrow k_1 = (*, \cdots)$$

on la tria del bit $*$ de k_1 es fa per majoria dels bits $*$ de $\mathbf{B}_1$ (els quals no han de ser necessàriament iguals)

Vist el procediment que fem servir haurem de demanar que L estigui suficientment limitat per disposar de suficients valors a $c_0(x)$ per tal d'omplir la taula $\mathbf{M}$. La tria dels L valors de cada columna s'haurà de fer de manera que es minimitzi la probabilitat d'error i finalment demanarem que sigui un nombre senar (i major que 1) per a què les decisions per majoria no puguin quedar en empat.

A continuació explicarem el procés que seguirem per obtenir una taula binària a partir de les imatges filtrades. Normalment $c_0(x)$ i $c_1(x)$ són taules de 128×128 i contenen valors complexos (ja que són funcions de variable complexa amb imatge també dins de $\mathbb{C}$). Per tal d'extreure'n una taula binària primer agafarem la porció central de 64×64 valors complexos, seguidament separarem la part real de la imaginària de tal manera que resulta una taula de 128×64 valors reals on les parts reals estan a les primeres 64 columnes i les parts imaginàries estan a les següents 64 columnes tal i com mostra el següent esquema.

$$c_0(x) = \begin{pmatrix} & \boxed{64 \times 64} & \end{pmatrix} \Rightarrow \begin{pmatrix} a_{0,0} + ib_{0,0} & \cdots & a_{0,63} + b_{0,63} \\ a_{1,0} + ib_{1,0} & & \\ \vdots & & \\ a_{63,0} + ib_{63,0} & & \end{pmatrix} \Rightarrow \cdots$$

$$\cdots \Rightarrow \left(\begin{array}{ccc|ccc} a_{0,0} & \cdots & a_{0,63} & b_{0,0} & \cdots & b_{0,63} \\ a_{1,0} & & & b_{1,0} & & \\ \vdots & & & \vdots & & \\ a_{63,0} & & & b_{63,0} & & \end{array}\right) \Rightarrow \mathbf{B}_0 = \begin{pmatrix} 0 & 1 & \cdots \\ 1 & 1 & \\ \vdots & & \end{pmatrix}$$

Finalment, es transforma aquesta taula de valors reals en una taula de valors binaris prenent la següent norma: si el valor de la taula de valors reals és negatiu, el bit corresponent serà un 0, si el valor és positiu o 0 el bit corresponent serà un 1.

Un cop obtinguda la clau k_1 hi haurà un procés de validació, explicat en la secció següent per comprovar si realment és la mateixa k_0 inicial o no. Si en la primera comprovació no obtenim una resposta afirmativa podem intentar desplaçar la porció que prenem de la imatge $c_1(x)$ per descartar errors que s'hagin pogut produir en la recuperació d'aquesta. En una imatge de 128×128 una cerca de ± 16 píxels per dimensió és suficient.

3.4 Construcció teòrica del sistema de la Biometric encryption

Amb els dos procediments explicats anteriorment tenim els elements necessaris per poder construir el nostre sistema de la *Biometric encryption*. Malgrat això, ens queda per aclarir un punt comentat anteriorment; un cop hem recuperat la clau k_1 necessitem comprovar que és la mateixa que la k_0 usada en un principi però ens trobem amb el fet que aquesta clau inicial ha estat descartada per garantir la seguretat del sistema, per tant no tenim manera de saber si la clau obtinguda servirà per desxifrar els nostres missatges i/o fitxers. Així doncs, en la fase d'inscripció abans de descartar la clau inicial k_0 la farem servir per xifrar S bits de dades. És convenient que aquests S bits siguin diferents en cada usuari per garantir la màxima seguretat. Per tant prendrem S bits de la part del filtre que emmagatzemem al *Bioscrypt*, és a dir d'H_{stored} i les xifrarem amb un criptosistema de clau privada (prenent com a clau k_0), a aquestes dades xifrades els aplicarem una funció de hash que ens retornarà el valor id_0 que també passarà a formar part del *Bioscrypt*.

Durant la fase de verificació, un cop recuperada k_1, agafarem els mateixos S bits de H_{stored} que hem usat en la fase prèvia i els xifrarem amb la nova clau criptogràfica k_1. Seguidament aplicarem la funció de hash a aquestes dades xifrades i obtindrem id_1. Si $id_0 = id_1$ llavors $k_0 = k_1$ amb probabilitat molt alta. A més a més l'identificador id_0 no conté informació sobre k_0 i per tant, pot ser emmagatzemat al *Bioscrypt*.

En les Figures 3.1 i 3.2 presentem un esquema del funcionament de les fases d'inscripció i verificació del sistema de la *Biometrc encryption*.

Com podem observar, finalment el *Bioscrypt* està format per l'H_{stored}, per la taula de coordenades $\mathbf{M}$ i l'identificador id_0 que ens permetrà legitimar la clau obtinguda en la fase de verificació k_1 a partir del nou senyal biomètric.

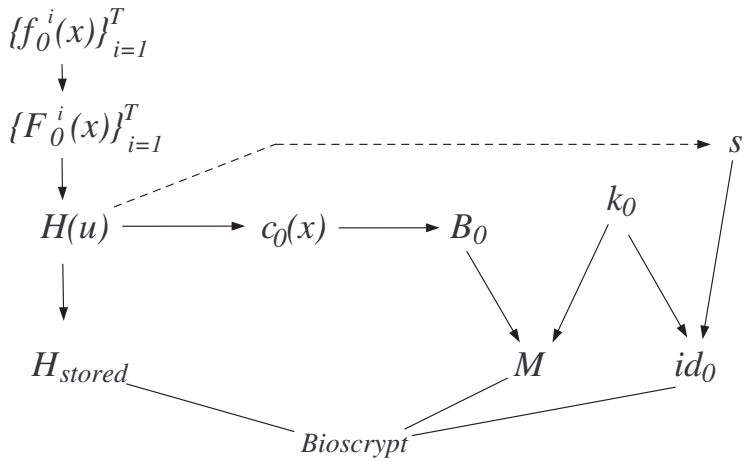


Figura 3.1: Esquema de la fase d'inscripció

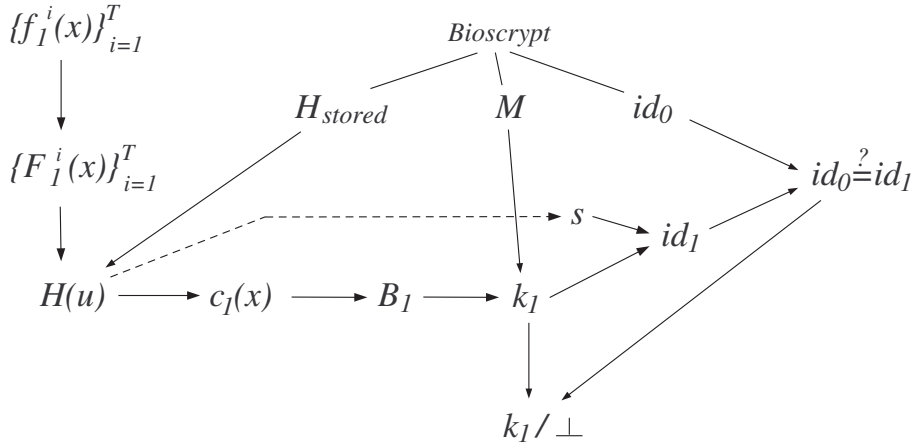


Figura 3.2: Esquema de la fase de verificació

3.5 Avantatges i inconvenients de la Biometric encryption

Tot i que aquest sistema i els *fuzzy extractors* són molt semblants, pel que fa a procediments podem trobar lleugeres diferències. Aquestes són precisament les que fan que puguem detectar certs avantatges i inconvenients respecte els dos protocols.

Els avantatges de la *Biometric encryption* són principalment, els deguts al major desenvolupament tècnic del sistema. És a dir, aquest protocol està molt més desenvolupat que els *fuzzy extractors* ja que incorpora un sistema de tractament d'imatge que, a part d'eliminar tots els problemes de coherència de mètriques que hem explicat en l'apartat dels *fuzzy extractors*, fan que el sistema sigui aplicable a qualsevol tipus de senyal biomètric tot i que potser s'haurien d'ajustar el nombre d'imatges de prova T i el coeficient de

tolerància del filtre α. No obstant això, trobem uns quants inconvenients que els *fuzzy extractors* resolen i que, en canvi, la *Biometric encryption* no. En [12] s'enumeren les diverses mancances de seguretat que presenta aquest sistema.

Un dels inconvenients principals de la *Biometric encryption* és la falta de robustesa és a dir, no hi ha cap garantia de seguretat davant de possibles modificacions en el *Bioscrypt*. Si un adversari modifica la informació pública emmagatzemada no està controlada la repercussió que aquest fet podria tenir en la recuperació de k_1 i a més a més no tenim cap garantia de poder detectar aquest tipus d'atac.

En canvi, en els *fuzzy extractors* robustos aquest tipus d'atacs estan perfectament controlats i són detectables. A més a més, com hem vist , en les últimes modificacions d'aquest tipus de *fuzzy extractors* la robustesa no implica una gran reducció de la clau extreta, cosa que en les primeres versions sí que passava.

Un altre inconvenient és degut al fet que la clau k_0, està lligada a posteriori amb el senyal biomètric. En canvi, en els *fuzzy extractors* la clau criptogràfica és extreta directament del senyal biomètric. Això fa que els *fuzzy extractors* siguin més segurs.

En definitiva podríem concloure que, a nivell tecnològic, la *Biometric encryption* està molt més desenvolupada i preparada per sortir al mercat mentre que els *fuzzy extractors* estan més avançats en el sentit algebraic i de seguretat però s'haurien d'introduir algunes ampliacions per poder ser utilitzats en el mercat d'avui dia.

Capítol 4

Millores

4.1 Millores generals

En aquesta secció plantejarem possibles millores tant per als *fuzzy extractors* com per a la *Biometric encryption*. Intentarem cobrir les mancances d'ambdós sistemes i incorporar noves idees.

En algunes de les seccions només plantejarem els problemes que hi ha i indicarem quins passos caldria seguir per acabar trobant una millora fent si cal un estudi de les possibles solucions. També exposarem algunes solucions que no seran factibles i n'explicarem les raons. En altres seccions trobarem alguna millora i especificarem com s'hauria d'implementar en el sistema per a què fos efectiva.

Tot i que durant les seccions anteriors hem enumerat els avantatges i inconvenients de cada sistema, a continuació elaborarem un llistat de possibles solucions a les mancances de cadascun.
En els *fuzzy extractors* robustos podríem millorar els aspectes següents:

- Introduir un procediment de tractament de la imatge per aconseguir un algorisme més concret.
- Solucionar el problema de la recuperació de la ω (sent ω el propi senyal biomètric o informació que el comprometria) durant la fase de verificació.

En la *Biometric encryption*, podríem millorar els aspectes següents:

- Millorar el codi de repetició usat durant el procés de combinar el senyal biomètric amb la clau criptogràfica.
- Intentar proveir de robustesa aquest sistema.

- Fer servir un altre sistema de tractament de la imatge més específic segons el tipus de senyal biomètric que estiguem tractant.

De tots aquests possibles temes a tractar, ometrem l'últim punt esmentat referent a la *Biometric encryption* ja que s'allunya bastant del tema central del projecte i es centra més en aspectes purament biomètrics.

Finalment també plantejarem el problema d'intentar trobar un híbrid entre els dos sistemes. Tot i que algunes de les qüestions que plantegem i de les seves possibles solucions van encaminades en aquesta direcció, tractarem aquest problema en una secció a part.

4.2 Millores dels fuzzy extractors

4.2.1 Tractament de la imatge

Les millores per implementar una fase de tractament de la imatge presenten molts problemes de coherència de mètriques ja que en els *fuzzy extractors* estudiats pràcticament només es treballa amb la distància de Hamming. L'estudi amb altres distàncies només es troba detallat en [5], ja que en la majoria d'articles es redueix a una petita secció al final.

Aquest fet ens porta a pensar en un tractament de la imatge coherent amb aquesta mètrica. Per exemple, el sistema explicat anteriorment en el capítol dels *fuzzy extractors* consistent en llegir la imatge bit a bit no seria coherent amb la distància de Hamming (no acceptaria, per exemple, errors deguts a translacions).

Si implementéssim un altre sistema com per exemple les minúcies en el cas de les empremtes dactilars, el primer problema que hauríem de solucionar seria com emmagatzemem la informació de les minúcies. Com hem explicat en seccions anteriors les minúcies es quantifiquen amb les coordenades (x, y) i amb la inclinació α de la singularitat. Així doncs, per treballar amb aquest tractament de la imatge podríem usar vectors que expressin, en les dues primeres posicions, les coordenades (x, y) de la minúcia i en la tercera posició l'angle α d'aquesta . Aquest sistema tampoc seria coherent amb la distància de Hamming ja que les dues primeres posicions pertanyen a una mètrica en $\mathbb{R}$ i l' última a una mètrica sobre $\mathbb{S}^1$. Per tant podríem intentar pensar el problema a la inversa i canviar la mètrica per implementar-la dins del sistema per a què tot funcionés correctament, però no sembla una tasca senzilla perquè s'haurien d'idear noves implementacions per a alguns procediments com per exemple, el *secure sketch*.

Així doncs, hem vist que no ens serà fàcil ni trobar una mètrica adient per a un sistema en concret, ni trobar un sistema adient per a la mètrica de Hamming. En seccions posteriors aquest problema ens tornarà a aparèixer.

4.2.2 Recuperació d'ω

El problema que ens plantegem en aquesta secció és el de la recuperació de l'input dels *fuzzy extractors* enmig del procés de validació.

De fet, l'input dels fuzzy extractors és ω però no s'especifica si aquest valor és el propi senyal biomètric o una codificació d'aquest.

Si donat ω es pot recuperar fàcilment el senyal biomètric llavors, si permetem a l'adversari veure el procés de validació, aquest podria interceptar el nostre senyal biomètric. Per tant, la solució més ràpida seria que ω fos el senyal biomètric xifrat d'alguna manera que no permetés recuperar el senyal en cas que l'adversari interceptés ω.
Aquesta solució presenta un problema greu ja que si anomenem f a la funció de xifrat i formalitzem les propietats que li demanem ens queda el següent:

Volem una funció $\omega = f(x; i)$ tal que coneguts els valors d' ω i de i (que representa els paràmetres del sistema) no puguem recuperar x (que és el senyal biomètric original).
Però aquesta funció també hauria de complir certes propietats en l'espai mètric en el qual treballem per a què el fuzzy extractor funcioni correctament. Per tant:

Volem una funció $\omega = f(x; i)$ tal que si $\mathbf{dis}(x, x') \leq \delta$ llavors $\mathbf{dis}(f(x; i), f(x'; i)) \leq \varepsilon$.
Fixem-nos que les dues propietats que volem són bastant oposades perquè l'una demana privacitat però l'altra demana que les imatges de valors propers siguin properes i per tant, ens estan facilitant informació sobre els valors que volíem mantenir en secret. A més a més aquesta funció haurà de ser injectiva és a dir, dos usuaris diferents no poden tenir la mateixa ω perquè en aquest cas la clau obtinguda pel *fuzzy extractor* en ambdós casos seria la mateixa i es comprometria la seguretat del sistema (un usuari podria per exemple, desxifrar la informació d'un altre).

Per tant podem concloure que aquest problema té difícil solució i que potser hauríem d'optar per alguna altra tècnica com ara protegir millor el procés de verificació en front dels adversaris per tal que ω sigui molt difícil d'interceptar.

4.3 Millores de la Biometric encryption

4.3.1 Millorar el codi

Una de les millores més clares a introduir, en un principi, és usar un altre codi que no sigui el de repetició en el *link algorithm*.

Fixem-nos que, quan usem aquest tipus de codi, les columnes de la matriu $\mathbf{M}$ ens estan dient que, en totes les coordenades d'una mateixa columna trobem el mateix bit en la imatge filtrada i binaritzada $\mathbf{B_0}$. Per tant, $\mathbf{M}$ ens està proporcionant informació de $\mathbf{B_0}$ i a priori, no sabem fins a quin punt pot ser utilitzada per un adversari per tal d'obtenir

patrons d'informació de $\mathbf{B_0}$.

Aquest problema es solucionaria considerablement si féssim servir un altre codi que no fos el de repetició ja que l'obtenció d'informació no seria tanta ni tan òbvia. En aquest context hauríem d'usar un codi corrector d'errors binari (no ens serviria, per exemple un codi de Reed Solomon ja que precisa treballar amb cossos de cardinal més gran).

El procediment a seguir serà codificar la clau mitjançant aquest codi i obtenir $cb_0 = CB(k_0)$.

$$k_0 = \overbrace{(0,1,\cdots,0)}^{128} \longrightarrow cb_0 = CB(k_0) = \overbrace{(1,0,0,\cdots,1)}^{n}$$

Un cop tinguem el valor cb_0, per a cada bit triarem (de manera aleatòria) unes coordenades de la imatge filtrada i binaritzada $\mathbf{B_0}$ que tinguin aquest bit. D'aquesta manera aconseguirem crear un vector $\mathbf{M}$ (o matriu depenent de com disposem la informació) de coordenades corresponents als bits de la clau k_0 codificada.

$$\mathbf{B}_0 = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \end{pmatrix}$$

$$\searrow$$

$$cb_0 = (1,0,0,\cdots,1) \rightarrow \mathbf{M} = ((x1,y1),(x2,y2),\cdots,(xn,yn))$$

Fixem-nos que ara $\mathbf{M}$ no ens fixa cap patró a priori ja que per a cada posició de la matriu tenim una sola coordenada (que pot correspondre a un bit 0 o 1). Segurament un adversari podria dur a terme estudis sobre el codi usat per intentar extreure informació de la clau o de $\mathbf{B_0}$ a partir d'$\mathbf{M}$ però en cap cas seria d'una manera tan senzilla com passava amb el codi de repetició. Per exemple podríem pensar que l'adversari obtindria informació si es repetissin certs patrons en la codificació de les claus (ja que el codi usat serà públic), però la clau codificada cb_0 així com $\mathbf{B_0}$ es descarten després de la seva utilització i, encara que certs usuaris tinguessin certs patrons repetits i les $\mathbf{B_0}$ corresponents coincidissin (fixem-nos que dos usuaris amb imatges diferents podrien tenir la mateixa matriu binaritzada ja que l'aplicació no és bijectiva), com que la tria de les coordenades de $\mathbf{B_0}$ que farem servir per crear $\mathbf{M}$ és un procés aleatori, el vector (o matriu) resultant serà totalment diferent. Per tant aquesta estratègia no serviria per obtenir informació.

Per a què tot aquest procés sigui segur i funcioni tal i com hem descrit, la matriu $\mathbf{B_0}$ ha de permetre fer la tria aleatòria de les coordenades per crear $\mathbf{M}$. Sabem que la matriu $\mathbf{B_0}$ conté $64 \times 64 = 4096$ bits, així doncs haurem d'adaptar la mida del vector/matriu $\mathbf{M}$ és a dir, la mida de cb_0 per tal que aquesta tria es pugui dur a terme.

En tot aquest procés estem usant un mètode *one time pad* ja que podem considerar que estem codificant el missatge cb_0 amb una clau $\mathbf{B_0}$ no aleatòria ni única per a cada usuari però sí molt més gran que el missatge que volem codificar. Aquest fet és el que ens

garantirà la seguretat del procés.

En definitiva l'objectiu serà buscar codis binaris correctors d'errors $[n, k, d]$ amb valors de dimensió adequats per tal que puguem garantir la seguretat. Estudiarem els codis següents:

- Codis de Repetició
- Codis de Hamming
- Codis de Reed-Muller
- Codis de Hadamard
- Codis de Golay
- Codis Simplex
- Combinacions de codis

D'aquests codis en farem un breu estudi veient les característiques que ens interessen de cadascun d'ells sense entrar en els detalls de cada codi (per a més informació sobre aquest i altres codis consulteu [10]). També farem un estudi de dues fites per veure a quins resultats teòrics òptims podríem arribar.

Codis de Repetició

Primer de tot ens centrarem en l'estudi del codi que es fa servir en la *Biometric encryption*, és a dir, el codi de repetició. L'objectiu és entendre quins paràmetres es tenen en compte i quins es són susceptibles de millora. Aquest codi consisteix en codificar cada bit repetint-lo L vegades.

$$0 \rightarrow \overbrace{00\cdots0}^{L} \qquad 1 \rightarrow \overbrace{11\cdots1}^{L}$$

Aquest codi permet corregir $t = \lfloor \frac{n-1}{2} \rfloor$ errors on n és la llargada del bit xifrat (en el nostre cas L). Aquest valor t s'anomena capacitat correctora del codi. De totes maneres pensem ara en la matriu $\mathbf{M}$ abans de substituir les seves entrades per coordenades tal i com mostra el següent esquema per a $L = 3$:

$$k_0 = (0, 0, 1, \cdots, 1)$$
$$\downarrow$$
$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 1 & & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 1 & & 1 \end{pmatrix}$$

Fixem-nos que, tot i que la capacitat correctora del codi és t (és a dir, podem corregir sempre t errors), si els bits erronis estan disposats de la manera adequada, en podem arribar a corregir més de t. De fet podríem arribar a corregir $128 \times t$ errors si estiguessin col·locats de manera òptima. Vegem el següent esquema on les posicions que contenen bits erronis estan indicades amb un $*$.

$$\mathbf{M} = \begin{pmatrix} * & 0 & 1 & & 1 \\ * & 0 & 1 & \cdots & 1 \\ 0 & 0 & 1 & & 1 \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} * & * & * & & * \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 1 & & 1 \end{pmatrix}$$

$$\downarrow \qquad\qquad \downarrow$$

$$k_1 = (*, 0, 1, \cdots, 1) \qquad k_1 = (0, 0, 1, \cdots, 1)$$

Veiem doncs que en el primer esquema dos bits erronis són suficients per provocar una recuperació errònia de la clau mentre que, en el segon esquema tot i que tenim més bits erronis podem recuperar la clau sense cap error. Veient aquesta propietat, quan realitzem l'estudi del codi, a part de tenir em compte la capacitat correctora d'un codi podem fixar-nos en el nombre de bits erronis que podríem arribar a corregir, com a màxim, si aquests estiguessin disposats de manera idònia. Amb aquests càlculs podem fer estudis reflectits en la Figura 4.1 i en la Taula 4.1 que conté els resultats més remarcables. En aquests resultats s'ha de tenir en compte que els codis de repetició amb L parell són pitjors que els que tenen L senar ja que en aquest últim cas, en la descodificació no ens podem trobar situacions d'empat.

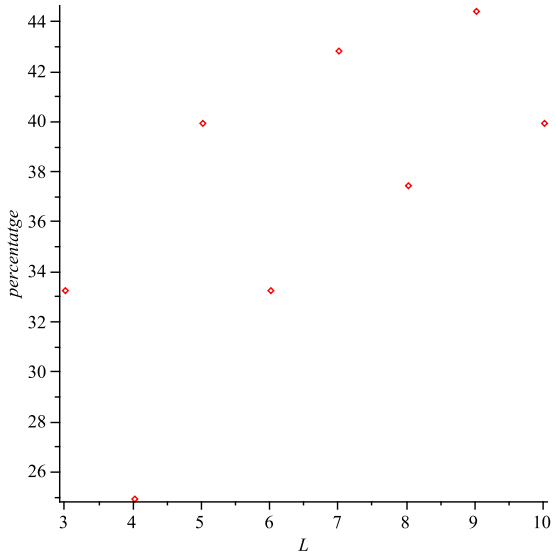


Figura 4.1: Percentatge màxim d'errors en funció de la dimensió del codi L

L	**bits totals**	**capacitat correctora**	**% màxim d'errors**
3	384	1	33,33 %
5	640	2	40 %
7	896	3	42,86 %

Taula 4.1: Resultats obtinguts amb el codi de repetició

Veiem doncs que, tot i que el percentatge màxim d'errors corregits (si aquests estan disposats de manera adequada) és elevat i augmenta amb L, la capacitat correctora és molt petita i creix molt lentament. Per altra banda, sabem que cada bit codificat (és a dir cada bit de la matriu $\mathbf{M}$) serà emparellat amb unes coordenades d'un bit de la matriu $\mathbf{B_0}$. Per tant ens haurem d'ajustar a aquestes dimensions. Com ja hem vist disposem d'una matriu $\mathbf{B_0}$ (porció central binaritzada de $c_0(x)$) de dimensions $64 \times 64 = 4096$ bits. Per garantir que la tria aleatòria es pugui dur a terme correctament hauríem d'agafar com menys bits millor i és per aquesta raó que només hem arribat fins a $L = 7$ en l'estudi d'aquest codi.

Codis de Hamming

Aquests codis són dels més coneguts en teoria de codis, van ser descoberts independentment per Marcel Golay el 1949 i Richard Hamming el 1950. Són perfectes i fàcils de descodificar. Quan els prenem binaris són equivalents a codis cíclics i tenen els paràmetres:

$$\mathcal{H}_2(r) = [n, k, d] = [2^r - 1, 2^r - 1 - r, 3]$$

En aquest cas estudiarem els casos per a $r = 3 \dots 7$ (ja que a partir de $r = 8$ ens trobem que $k > 128$ i la clau que fem servir només té 128 bits). En cada cas estudiat el nostre objectiu serà separar la clau k_0 en tants blocs com necessitem i codificar cada bloc amb el codi que estiguem tractant (de fet aquest mètode ja l'usàvem amb el codi de repetició amb blocs d'un bit). La capacitat correctora del codi en tots els casos serà $t = \lfloor \frac{3-1}{2} \rfloor = 1$ per tant, només ens fixarem en el % màxim d'errors que podrem arribar a corregir (de cada bloc en podríem corregir un per tant, si els errors estan disposats adequadament podríem arribar a corregir tants errors com blocs). En cas que la llargada dels blocs no sigui divisible per la llargada de la clau, sempre prendrem la part entera superior de la divisió (suposant si cal, que codifiquem més de 128 bits). Vegem els resultats obtinguts estudiant els diferents codis de Hamming en la Figura 4.2 i la Taula 4.2.

Així doncs, veiem que tot i que aquest tipus de codis utilitza molt pocs bits, no són gens recomanables per al nostre objectiu perquè tenen tant la capacitat correctora com el % d'errors molt baixos.

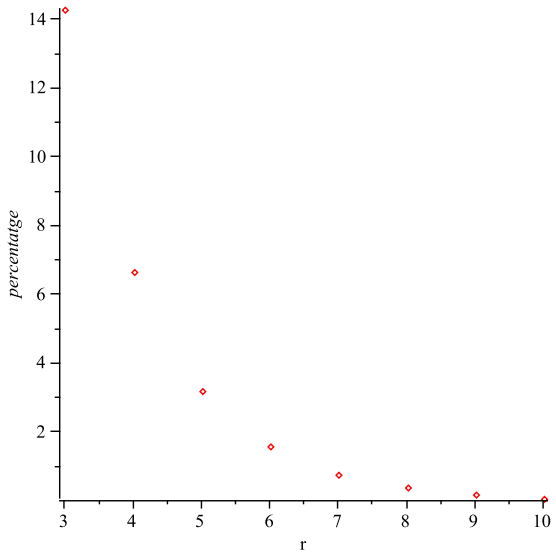


Figura 4.2: Percentatge màxim d'errors en funció de la r del codi

r	**codi**	**nombre de blocs**	**bits totals**	**% màxim d'errors**
3	[7, 4,3]	32	224	14,29 %
4	[15,11,3]	12	180	6,67%
5	[31,26,3]	5	155	3,23%
6	[63,57,3]	3	189	1,59%
7	[127,120,3]	2	254	0,79 %

Taula 4.2: Resultats obtinguts amb el codi de Hamming

Codis de Reed-Muller

Els codis de Reed-Muller són codis binaris i tenen bones propietats a l'hora de descodificar-se. El codi $\mathcal{R}(1,5)$ va ser usat per la sonda Mariner 9 per transmetre imatges en blanc i negre de Mart el 1972. Per al cas general, aquests codis són del tipus:

$$\mathcal{R}(r,m) = [n,k,d] = [2^m, 1 + \binom{m}{1} + \ldots + \binom{m}{r}, 2^{m-r}]$$

Farem un estudi sobre tots els codis d'aquest tipus amb $m = 3 \ldots 10$ i $r = 1 \ldots m-2$ (no arribem a $r = m$ per poder aconseguir valors de distàncies interessants).

Com ja hem explicat, amb el codi de repetició es donava molta informació sobre la imatge biomètrica. En canvi aquesta vegada la taula no en dóna tanta si més no, a priori. Per tant podem treballar amb codis que obtinguin una cb_0 més gran si amb això aconseguim millors resultats.

A continuació presentem la Figura 4.3 amb els codis estudiats on els codis amb r diferent però amb la mateixa m estan representats amb el mateix color i posteriorment, la Taula 4.3 amb els resultats més remarcables. En aquesta última ens restringim a aquells en els quals la capacitat correctora sigui > 1, $k \leq 128$ i la clau codificada no ocupi més de 1500 bits aproximadament.

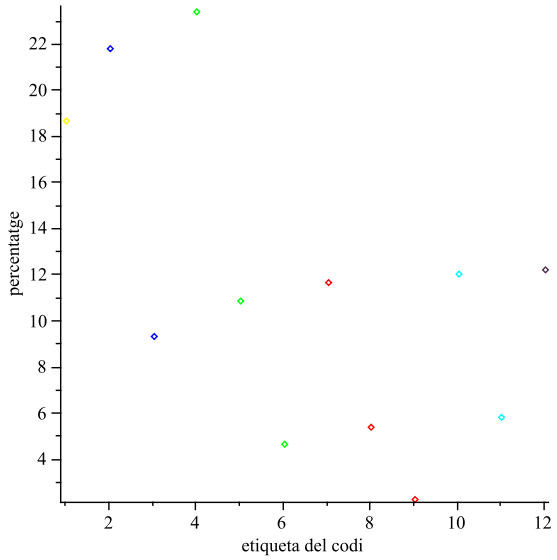


Figura 4.3: Percentatge màxim d'errors en funció del codi

etiqueta	**codi**	**blocs**	**bits totals**	**cap. correctora**	**% màxim d'errors**
1	$[16, 5, 8]$	26	416	3	18,75 %
2	$[32, 6, 16]$	22	704	7	21,875 %
3	$[32, 16, 8]$	8	256	3	9,375 %
4	$[64, 7, 32]$	19	1216	15	23,44 %
5	$[64, 22, 16]$	6	384	7	10,94 %
6	$[64, 42, 8]$	4	256	3	4,69 %
7	$[128, 29, 32]$	5	640	15	11,71 %
8	$[128, 64, 16]$	2	256	7	5,47%
9	$[128, 99, 8]$	2	256	3	2,34%
10	$[256, 37, 64]$	4	1024	31	12,11%
11	$[256, 93, 32]$	2	512	15	5,86%
12	$[512, 46, 128]$	3	1536	63	12,3%

Taula 4.3: Resultats obtinguts amb el codi de Reed-Muller

Com podem veure el codi amb més capacitat correctora és el $[512, 46, 128]$ i veiem

també que usa pocs blocs (fet que farà millorar el cost computacional). A més a més, si els errors estiguessin disposats de manera adequada podríem arribar a corregir un 12% dels 1536 bits de la clau codificada.

El codi amb un percentatge més elevat és el $[64, 7, 32]$ que corregeix 15 errors i usa 19 blocs. Aquest codi no corregeix tants errors com l'anterior però si estan ben disposats pot arribar a corregir fins a un 23% dels 1216 bits que fa servir.

Si volem un codi que no necessiti massa redundància podem fer servir el $[128, 29, 32]$ que ens passa de la clau k_0 de 128 bits a una paraula del codi de 640 bits i arriba a corregir 15 errors. Tanmateix els codis de Hamming eren els millors en aquest aspecte.

Codis de Hadamard

Aquests codis deuen el seu nom al matemàtic francès Jacques Hadamard i s'obtenen a partir de les matrius de Hadamard. Són codis binaris i tenen els paràmetres següents:

$$[n, k, d] = [n, \log_2(2n), \frac{n}{2}]$$

En la Figura 4.4 veiem un estudi dels codis de Hadamard sense tenir en compte que $k \in \mathbb{Z}$. Aquest fet però, sí que l'hem tingut en consideració per a l'estudi posterior dels casos concrets.

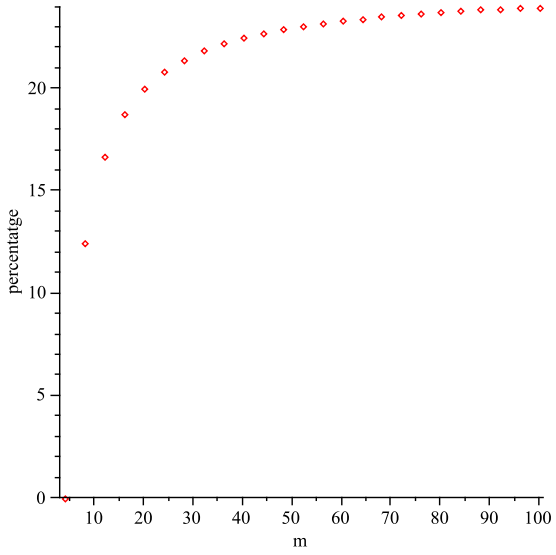


Figura 4.4: Percentatge màxim d'errors en funció de la m del codi

Els codis que resulten interessants d'estudiar d'aquest tipus resulten ser els següents:

$$[16, 5, 8], [32, 6, 16] \text{ i } [64, 7, 32]$$

Tots tres han estat estudiats en l'apartat anterior com a codis de Reed-Muller. Si intentéssim aconseguir un codi que fes servir un sol bloc és a dir, amb $k = 128$ llavors $n = 2^{127} \gg 4096$ i per tant no es podria dur a terme.

Codis de Golay

Aquests codis van ser introduïts per Marcel Golay el 1948. Es denoten per $\mathcal{G}_{23}$, $\mathcal{G}_{24}$, $\mathcal{G}_{11}$ i $\mathcal{G}_{12}$. Els dos primers són binaris i els altres dos ternaris. En particular, el $\mathcal{G}_{24}$ va ser usat per la sonda Voyager per transmetre imatges de Júpiter i Saturn en color l'any 1979. Els dos codis de Golay binaris tenen els següents paràmetres i permeten obtenir els resultats exposats en la Taula 4.4:

$$\mathcal{G}_{24} = [24, 12, 8] \quad \text{i} \quad \mathcal{G}_{23} = [23, 12, 8]$$

codi	**blocs**	**bits totals**	**cap. correctora**	**% màxim d'errors**
$[24, 12, 8]$	11	253	3	13,04 %
$[23, 12, 7]$	11	264	3	12,5 %

Taula 4.4: Resultats obtinguts amb els codis de Golay

Per tant només millorem els resultats aconseguits amb els codis de Reed Muller amb el fet que necessitem poca redundància per aconseguir aquests resultats. No obstant el nostre interès se centra més en aconseguir capacitats correctores altes encara que haguem de fer servir més redundància i és per això que aquests codis no són els més adequats en aquest cas.

Codis Simplex

Aquests codis són els duals dels codis de Hamming, és a dir, tenen per matriu generadora la matriu de paritat dels codis de Hamming. Aquests codis són binaris (evident ja que els codis de Hamming també ho eren) i tenen la forma:

$$[n, k, d] = [2^r - 1, r, 2^{r-1}]$$

Prenent valors de r des de 3 fins a 6 (ja que amb $r = 7$ la clau codificada ja ens ocuparia més de 2000 bits) obtenim la Figura 4.5 i la Taula 4.5.

Veiem que amb aquests codis s'aconsegueixen percentatges més alts que amb els de Reed-Muller però en canvi capacitats correctores més baixes.

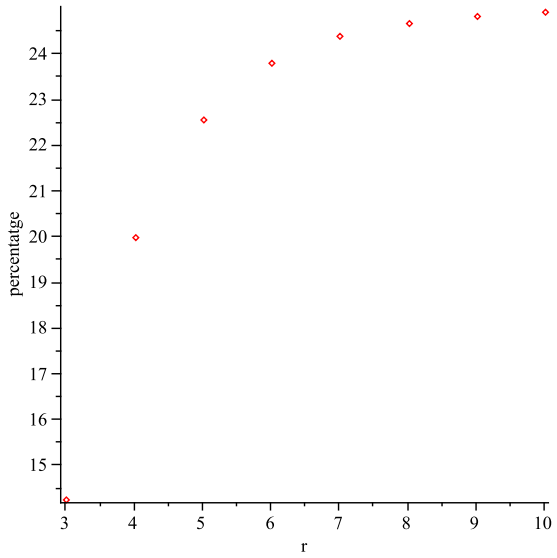


Figura 4.5: Percentatge màxim d'errors en funció de la r del codi

codi	**blocs**	**bits totals**	**cap. correctora**	**% màxim d'errors**
$[7, 3, 4]$	43	301	1	14,29 %
$[15, 4, 8]$	32	480	3	20 %
$[31, 5, 16]$	26	806	7	22,58 %
$[63, 6, 32]$	22	1386	15	23,81 %

Taula 4.5: Resultats obtinguts amb el codi de Simplex

Combinacions de codis: Hamming + Repetició

L'estratègia a seguir en aquesta secció és la combinació de dos codis per aconseguir un codi millor que els primers.

En aquest cas prendrem el codi de Hamming $[7, 4, 3]$ i el codi de repetició amb $L = 3$. L'estratègia a seguir serà la següent: prendrem la clau k_0 i la codificarem amb el codi de Hamming anterior obtenint $c_h(k_0)$ i, després d'això, codificarem aquest valor amb el codi de repetició obtenint finalment $c_r(c_h(k_0))$.

Aquest procés es pot entendre com un enviament de la paraula codificada amb Hamming 3 vegades per assegurar la comunicació. És a dir,

$$(a_1, a_2, a_3, a_4) \rightarrow (c_1, \ldots, c_7, c_1, \ldots c_7, c_1, \ldots, c_7)$$

on $(c_1, \ldots c_7)$ és la codificació del vector (a_1, a_2, a_3, a_4) amb el codi de Hamming.

Amb aquesta construcció podem arribar a corregir 4 errors ja que la distància de Hamming del codi resultant és $d = 3 * 3 = 9 \rightarrow t = 4$

El percentatge màxim d'errors que podríem arribar a corregir en cas que aquests estiguessin ben disposats l'obtindríem considerant que una de les paraules del codi de Hamming es transmet de manera totalment errònia, i permetent un error en alguna de les altres posicions restants, és a dir:

$$(\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_7, \tilde{c}_1, c_2, \ldots c_7, c_1, \ldots, c_7)$$

per tant el percentatge màxim que es podria arribar a corregir seria d'un 42,86%. És a dir, amb aquesta construcció recuperem el percentatge obtingut amb el codi de repetició amb $L = 7$ i augmentem un bit la capacitat correctora. A més a més, en aquest cas fem servir només 672 bits.

Aquest tipus de combinació de codis torna a donar informació a l'adversari sobre coordenades on apareix el mateix bit. De totes maneres en el cas del codi de repetició sabíem les coordenades de 7 bits iguals i, en el cas que tractem només sabem les coordenades de 3 bits iguals. A més a més aquests 3 bits corresponen a un bit de la clau codificada, mentre que els 7 bits pertanyen a un bit de la clau k_0.

Combinacions de codis: Golay + Repetició

Aquesta combinació implicarà el codi de Golay $[23, 12, 7]$ i novament el codi de repetició amb $L = 3$.

En aquest cas la construcció obtinguda pot corregir fins a 10 errors i el percentatge màxim de correcció en cas que els errors estiguin ben disposats és del 42,02%. És a dir que augmentem la capacitat correctora sense que el percentatge es vegi pràcticament afectat.

Malgrat això augmentem lleugerament el nombre de bits necessaris ja que en aquest cas en necessitem 759, però encara fem servir menys bits que en el cas del codi de repetició amb $L = 7$ en el qual en fèiem servir 896.

Fer servir el codi de repetició donarà de nou informació a l'adversari sobre coordenades on apareixen els mateixos bits.

Estudi de cotes

Veient els resultats obtinguts ens adonem que no és trivial trobar un codi millor que el de repetició en tots els aspectes ja que si es millora la capacitat correctora (que fa referència als errors que podem corregir en tots els casos) ens baixa molt el percentatge d'errors que podem corregir com a màxim en el cas que aquests estiguin posats de manera adequada.

De totes maneres si ens haguéssim de decantar per un dels dos valors evidentment escolliríem la capacitat correctora ja que és la que ens funciona per a totes les disposicions

possibles d'errors.

Abans d'exposar les conclusions farem un petit estudi amb dues cotes sobre teoria de codis: la cota de Singleton i la cota de Plotkin. Amb aquestes dues cotes buscarem quins paràmetres hauria de tenir un codi que només utilitzi un sol bloc, és a dir, que el percentatge màxim d'errors que es puguin corregir sigui igual a t/n on t és la capacitat correctora del codi.

La cota de Singleton ens diu que si tenim un codi corrector d'errors $[n, k, d]$ es compleix que:

$$d \leq n - k + 1$$

Per tant en el cas òptim (assolit per alguns codis com per exemple els de Reed Solomon) tenim que:

$$d = n - k + 1$$

Si volem trobar un codi (òptim) que faci servir un sol bloc (és a dir $k = 128$) i donem valors a la n entre 128 i 1000 (quedant determinada la distància òptima per l'equació anterior) trobarem la següent gràfica (Figura 4.6) que expressa el percentatge d'errors que podríem corregir en funció de la n que faci servir el codi.

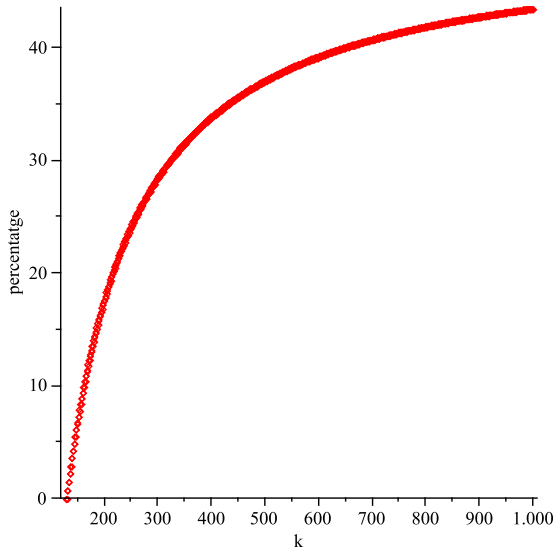


Figura 4.6: Percentatge màxim d'errors en funció de la k segons la cota de Singleton

Per tant veiem que en un principi sembla possible que per a un codi amb $k = 128$ i $n \approx 800$ es pugui aconseguir un percentatge de correcció proper al 50%.

Tanmateix, estudiem ara la cota de Plotkin, aquesta ens diu que si tenim un codi $[n, k, d]_q$ llavors:

$$d \leq \frac{nq^{k-1}(q-1)}{q^k - 1}$$

En el nostre cas $q = 2$ ja que tractem codis binaris i prenent el cas òptim tenim que:

$$d = \frac{n2^{k-1}}{2^k - 1}$$

I fent el mateix estudi que amb la cota de Singleton obtenim la següent gràfica (Figura 4.7).

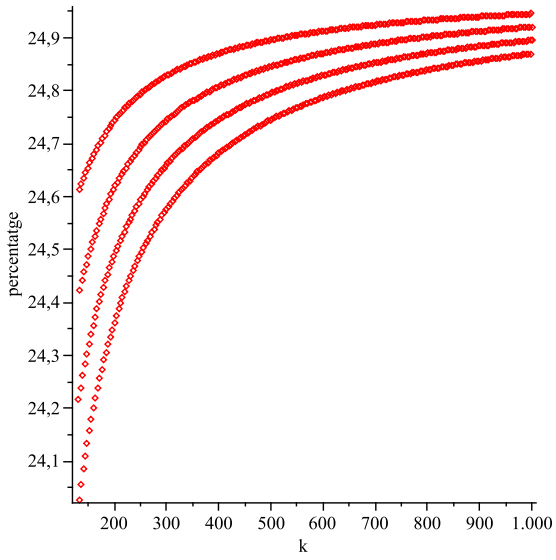


Figura 4.7: Percentatge màxim d'errors en funció de la k segons la cota de Plotkin

Per tant veiem que fent servir codis binaris i usant $k = 128$ (és a dir un sol bloc) i $n \approx 800$ bits podríem arribar (en el cas òptim) a una capacitat correctora igual a una quarta part de la n, és a dir a corregir $\approx$200 errors. De totes maneres veiem que en la majoria de casos que hem tractat, fer servir $k = 128$ implica n molt gran i en els pocs que això no passava, la capacitat correctora era molt petita. Per tant els resultats que hem obtingut disten força d'aquesta cota.

Conclusions

En tot aquest estudi s'ha de tenir en compte que no sabem l'entropia que presenten les empremtes dactilars i altres senyals biomètrics. A més a més, tampoc sabem si certes regions són més propenses a patir errors que d'altres. Per tant, no tenim manera de garantir que els errors estiguin ben distribuïts en cap cas. És a dir, l'estratègia de fer servir un codi de repetició no és l'òptima tenint en compte aquest raonament i, segurament seria molt millor usar algun altre codi com per exemple el de Reed-Muller [512, 46, 128] aconseguint així la correcció d'un nombre major d'errors, estiguin en la posició que estiguin, tot i que el tant per cent d'errors que podem corregir en cas que estiguin de manera adequada disminueixi considerablement.

De totes maneres si estem interessats en resultats semblants als que ja teníem amb el codi de repetició es poden usar les combinacions de codis estudiades que milloren lleugerament la capacitat correctora i els bits totals usats sense que es vegi afectat el percentatge d'errors.

També es podria intentar l'estratègia de fer servir alguna tècnica com l'*interleaving* per evitar ràfegues d'errors. Aquesta tècnica funcionaria si estiguéssim segurs que les dades poden patir amb probabilitat alta aquest tipus d'errors. Però com ja hem dit, no sabem quin tipus d'errors poden presentar les dades biomètriques.

4.3.2 Introduir robustesa

Per a la introducció d'aquesta propietat en el sistema de la *Biomectric Encryption* ens haurem d'assegurar que ningú pugui modificar el *Bioscrypt* i que, en cas que es modifiqui, es pugui detectar amb probabilitat alta.

Considerarem que l'adversari guanya si aconsegueix que es regeneri una clau diferent a la original sense que l'usuari legítim ho hagi pogut detectar.

En primer lloc farem un estudi dels problemes que comportaria la modificació dels paràmetres del *Bioscrypt* un a un.

- Si l'adversari només modifica l'identificador $id_0 \rightarrow \tilde{id}_0$, quan intentem recuperar la clau en la fase de verificació ens podem trobar amb dues situacions diferents. Si $k_1 = k_0$ llavors $id_0 = id_1 \neq \tilde{id}_0$ i el sistema detectaria l'error. Si en canvi degut als errors que puguin presentar les imatges, $k_1 \neq k_0$, llavors no hi ha cap garantia que $id_1 \neq \tilde{id}_0$ tot i que l'adversari no té cap poder sobre la probabilitat d'encert ja que els errors de la clau són deguts a factors externs (estem suposant que l'adversari no modifica res més).

- Si l'adversari només modifica la matriu $\mathbf{M}$ amb la finalitat de modificar la clau $k_1 \neq k_0$ llavors $id_1 \neq id_0$ i per tant detectaríem l'error.

- Si l'adversari només modifica l'H_{stored} llavors no només està influint en la recuperació de la imatge filtrada $c_1(x)$ i, per tant, de la clau k_1 sinó que també està influint en els s bits que xifrarem per recuperar l'identificador id_1. No tenim doncs cap garantia de seguretat ja que si xifrem els s bits d'H_{stored} amb k_0 i els $\tilde{s}$ bits d'$\tilde{H}_{stored}$ amb $k_1 \neq k_0$ llavors no sabem res sobre id_0 i id_1 (hauríem d'evitar la possibilitat que $id_0 = id_1$ amb $s \neq \tilde{s}$ i $k_0 \neq k_1$).

Així doncs podem concloure que l'únic valor que sempre seria detectable en front de modificacions externes és $\mathbf{M}$. Així doncs haurem de garantir la integritat de H_{stored} i id_0.

Per garantir aquesta propietat farem servir un KMS-MAC ja que haurem de permetre modificacions de la clau. Aquest tipus de funció, com ja hem vist en apartats anteriors, usa dos inputs per generar una etiqueta σ. Una de les entrades del KMS-MAC és pública mentre que l'altra és privada (o no accessible per a l'adversari).

En un primer moment podríem plantejar-nos la possibilitat d'introduir al KMS-MAC un conjunt format per H_{stored} i id_0 com a input públic i k_0 com a input privat. Però fixem-nos també si el KMS-MAC garanteix la integritat de k_0, llavors la funció de l'id_0 és totalment redundant.

Per tant eliminarem la fase en la qual es creava l'identificador id_0 i la substituirem per una fase on es crearà l'etiqueta del KMS-MAC. Podem agafar els mateixos s bits d'H_{stored} i usar-los d'input públic del KMS-MAC.

Fent això, les possibles modificacions de la informació que queda del *Bioscrypt* consistent en els bits restants d'H_{stored} i la matriu $\mathbf{M}$, només acabarien afectant a la recuperació de la clau k_0 (que farem servir com a input del KMS-MAC) i que, per tant, no es pot modificar sense que es detecti. És a dir, suposem que un adversari només modifica $\mathbf{M}$ i els bits d'H_{stored} que no farem servir en la generació de l'etiqueta. L'objectiu de l'adversari és modificar la clau recuperada i per això suposarem que, amb aquestes modificacions (amb les quals només podem tenir efecte sobre la recuperació de la clau), $k_1 \neq k_0$. Ara agafem els s bits (que no s'han modificat) d'H_{stored} i els introduïm al KMS-MAC juntament amb la nova clau k_1, evidentment la nova etiqueta $\tilde{\sigma}$ no serà igual a la inicial σ amb una probabilitat suficientment alta.

Així doncs, podem concloure que només cal garantir la integritat de k_0 i dels s bits d'H_{stored} per detectar possibles modificacions del *Bioscrypt* per part de l'adversari, el qual té la finalitat de modificar la clau sense ser detectat.

Implementarem doncs un KMS-MAC que agafarà els s bits d'H_{stored} i la k_0 i crearà un tercer valor σ:

$$\mathbf{MAC}(s, k_0) = \sigma$$

Aquest valor l'emmagatzemarem al *Bioscrypt*, per tant en aquesta nova versió del sistema de la *Biometric encryption* modificarem la fase de creació de l'identificador. En la fase d'inscripció usarem els s bits d'H_{stored} juntament amb la clau criptogràfica generada a l'atzar k_0 com a input del KMS-MAC. La resposta d'aquest serà σ i emmagatzemarà en el *Bioscrypt* que quedarà com un conjunt de tres elements: H_{stored}, $\mathbf{M}$ i σ. Durant la fase de verificació, en comptes de comprovar que els identificadors són iguals, és a dir $id_0 = id_1$, verificarem si $MAC(\tilde{s}, k_1) \stackrel{?}{=} \tilde{\sigma}$. En cas que aquesta comprovació sigui correcta acceptarem la clau regenerada k_1. En cas contrari la rebutjarem.

Inscripció	Verficació
$\mathbf{MAC}(s, k_0) = \sigma$	$\tilde{s}, \tilde{\sigma}, k_1 \rightarrow \mathbf{MAC}(\tilde{s}, k_1) \stackrel{?}{=} \tilde{\sigma}$
$\downarrow$	$\downarrow$
$Bioscrypt = (H_{stored}, \mathbf{M}, \sigma)$	sí $\rightarrow k_0 = k_1$
	no $\rightarrow \perp$

4.4 Creació d'un híbrid

Fins ara hem intentat buscar millores independents dels dos sistemes estudiats (*fuzzy extractors* i *Biometric encryption*) però en aquesta secció intentarem combinar-los per aconseguir les propietats bones d'ambdós sistemes.

Com ja sabem, l'avantatge principal del sistema de la *Biometric encryption* és la incorporació del tractament de la imatge. No obstant això, la resta del procés (la part algebraica) és millor en els *fuzzy extractors*. Per tant la creació d'un híbrid hauria d'intentar agafar la fase de tractat d'imatge de la *Biometric encryption* i usar-la com a entrada (és a dir com a ω) dels *fuzzy extractors*. Veurem però que aquest procés comporta molts problemes.

Fixem-nos que la fase de tractament de la imatge de la *Biometric encryption* i la posterior fase de binarització, ens permeten extreure a partir d'un escanejat del senyal biomètric una matriu binària $\mathbf{B_0}$.

Podríem plantejar-nos la possibilitat que aquesta matriu fos utilitzada com a entrada per als *fuzzy extractors* (és a dir, com a ω) però presenta uns quants problemes que detallem a continuació:

- El primer problema (i el principal) és que aquesta matriu no és única per a cada usuari, és a dir, fàcilment podem trobar dos usuaris que evidentment tindran $c_0(x)$ diferents però que, al dur a terme el procés de binarització, resultin $\mathbf{B_0}$ iguals. Aquest fet es podria intentar solucionar usant algun altre procés de binarització diferent, és

a dir, per a cada posició de $c_0(x)$ podríem emmagatzemar més d'un bit. D'aquesta manera disminuiríem les possibilitats de coincidència entre usuaris tot i que no les eliminaríem del tot.

- No sabem quina entropia presenten les dades de $\mathbf{B_0}$, és a dir encara que les dades inicials tinguin una entropia donada $\geq m$, no sabem quina és l'entropia de les $\mathbf{B_0}$ i les entrades dels *fuzzy extractors* requereixen un cert llindar d'entropia. De fet estem usant una entrada amb entropia desconeguda ($\mathbf{B_0}$) extreta a partir d'unes dades que no són aleatòries (les f_0^i). La diferència està en el fet que en la *Biometric encryption* usem aquesta entrada per xifrar una nova clau k_0 que sí que és totalment aleatòria. En canvi en els *fuzzy extractors*, d'aquesta $\mathbf{B_0}$ n'hem d'extreure una clau aleatòria que es farà servir per xifrar i desxifrar missatges.

- Els dos problemes anteriors no s'ajuden mútuament ja que si no sabem quina entropia conté $\mathbf{B_0}$ ni com té distribuïda la informació, encara que augmentem els bits per valor de $c_0(x)$ podria passar que hi haguessin certes zones del senyal biomètric molt semblant per a tots els usuaris. Aquest fet faria tornar a augmentar les possibilitats de coincidència entre dos o més usuaris.

Vistos tots aquests inconvenients podem plantejar-nos la possibilitat d'augmentar els bits per valor $c_0(x)$ al màxim, és a dir, fer servir la pròpia matriu $c_0(x)$ com a entrada dels *fuzzy extractors*. Aquest fet comportaria un dels inconvenients que ja teníem amb $\mathbf{B_0}$, no sabem quina entropia té $c_0(x)$. Si suposem que les imatges inicials tenen una entropia donada $\geq m$, en fer la transformada de Fourier mantindríem aquesta entropia (ja que estem usant una funció bijectiva) però després usem un procés de filtrat que incorpora aleatorietat (que en principi faria augmentar el grau d'entropia) i fem una mitjana de totes les dades (que faria disminuir l'entropia). Amb tot aquest procés, no sabem quina entropia ens queda i per tant, s'hauria de procedir a un estudi més detallat per veure si aquesta opció és viable o no.

De totes maneres, encara que l'entropia estigués dins dels límits vàlids, ens trobaríem amb altres problemes, com per exemple el fet que no sabem el rang de valors que pot arribar a tenir $c_0(x)$, per tant la binarització podria arribar a ocupar molta memòria.

Si no binaritzéssim $c_0(x)$ i féssim servir una altra mètrica per poder tractar amb aquest valor com a entrada dels *fuzzy extractors* ens trobaríem amb el problema de la coherència mètrica explicada en seccions anteriors ja que tots els procediments estudiats amb els *fuzzy extractors* funcionen amb la mètrica de Hamming.

A més a més, encara que aquesta solució fos viable, no arreglaria el problema de la recuperació d'ω enmig del procés de validació dels *fuzzy extractors*. De fet, $\mathbf{B_0}$ sí que arreglaria aquest problema ja que és una funció que donats dos valors propers la matriu

és semblant i, a més a més, a partir d'aquesta $\mathbf{B_0}$ no ens és possible recuperar les dades biomètriques (sense un nou escanejat i el filtre H_{stored} emmagatzemat). Però la raó per la qual aquest valor és vàlid es degut al fet que no es tracta d'una aplicació bijectiva i per tant, no podem tornar enrera. I aquesta propietat comporta la no unicitat entre usuaris, fet que fa impracticable usar $\mathbf{B_0}$ com a input.

Però fixem-nos que si passem a l'altre extrem fent servir tots els bits possibles, és a dir usem $c_0(x)$, llavors aquesta informació (que correspondria a la ω dels *fuzzy extractors*) juntament amb H_{stored} ens proporcionaria informació sobre el nostre senyal biomètric. Fixem-nos que, si $C_0(x)$ és la transformada de Fourier de $c_0(x)$:

$$H_{stored} = e^{-i\phi_{A_0}} e^{i\phi_R}$$

$$C_0(u) = A_0(u) \frac{|A_0(u)| e^{-i\phi_{A_0}}}{\alpha P(u) + \sqrt{1-\alpha^2} D_0(u)} e^{i\phi_R}$$

Per tant:

$$\frac{C_0(u)}{H_{stored}} = A_0(u) \frac{|A_0(u)|}{\alpha P(u) + \sqrt{1-\alpha^2} D_0(u)}$$

que és una funció que només depèn dels escanejats inicials (no conté aleatorietat i, per tant, pot donar informació a l'adversari sobre el nostre senyal biomètric).

Veiem doncs que poder crear un híbrid no és gens trivial i que comporta molts problemes (molts més dels que ens podríem pensar en un principi).

De fet, en les seccions anteriors, el que estàvem fent (introduint robustesa i millorant el codi de repetició) era intentar acostar la *Biometric encryption* als *fuzzy extractors*. És a dir, que en certa manera ja estàvem intentant crear una espècie d'híbrid entre els dos sistemes.

Capítol 5

Codis detectors de manipulacions algebraiques

5.1 Introducció

En aquest capítol parlarem dels codis detectors de manipulacions algebraiques (AMD-*codes*) i de la seva importància a l'hora de construir codis d'autentificació de missatges amb seguretat contra manipulacions de clau (KMS-MAC).

Ens situarem en un context on tenim un dispositiu d'emmagatzematge abstracte $\Sigma(\mathcal{G})$ que pot contenir un sol element $x \in \mathcal{G}$. Aquest dispositiu és privat en el sentit que l'adversari no té poder per llegir totalment el contingut de $\Sigma(\mathcal{G})$. Tanmateix, pot manipular el valor emmagatzemat x afegint-hi un excés $\Delta \in \mathcal{G}$ de manera que $\Sigma(\mathcal{G})$ passi a contenir $x + \Delta \in \mathcal{G}$. El nostre objectiu és fer servir $\Sigma(\mathcal{G})$ per autentificar missatges.

Per això introduïm la noció de AMD-*code* que xifra un missatge s donant com a resultat x que s'emmagatzemarà a $\Sigma(\mathcal{G})$. Qualsevol modificació d'aquest valor es detectarà amb probabilitat d'error $\leq \delta$.

També parlarem de KMS-MAC que, donat un missatge s produirà, amb l'ajut d'una clau $k \in \mathcal{G}$, una etiqueta σ. La parella (s, σ) es guardarà en un dispositiu públic (i, per tant, no segur) mentre que k s'emmagatzemarà en el dispositiu privat $\Sigma(\mathcal{G})$. El nostre objectiu serà que qualsevol adversari havent vist (s, σ), sigui incapaç de modificar la clau $k \to k + \Delta \in \mathcal{G}$ de manera que la nova clau modificada sigui la corresponent a una nova parella $(\tilde{s}, \tilde{\sigma})$ amb una probabilitat d'error $\leq \delta$.

Els AMD-*codes* tenen múltiples aplicacions, com per exemple els codis d'autentificació de missatges amb seguretat contra manipulacions de clau (KMS-MACs), que hem fet servir en la construcció dels *fuzzy extractors* robustos, i també veurem una aplicació per convertir qualsevol esquema lineal de compartició de secrets (*secret sharing*) en un esquema robust

que ens asseguraria, en aquest cas, que cap conjunt amb les característiques adequades de participants corruptes pogués forçar la recuperació d'un secret diferent a l'original sense que els participants honestos ho poguessin detectar.

5.2 Definicions

Un (S, G, δ)-codi detector de manipulacions algebraiques (AMD-*code*) és una funció probabilística de codificació $\mathcal{E} : \mathcal{S} \to \mathcal{G}$ d'un conjunt $\mathcal{S}$ de mida S en un grup $\mathcal{G}$ d'ordre G juntament amb una funció per descodificar $\mathcal{D} : \mathcal{G} \to \mathcal{S} \cup \{\perp\}$ tal que $\mathcal{D}(\mathcal{E}(s)) = s$ amb probabilitat 1 per a qualsevol $s \in \mathcal{S}$.

Hi ha dos tipus de AMD-*codes* depenent del grau de seguretat que vulguem aconseguir. Així doncs tenim els AMD-*codes* amb seguretat dèbil i els AMD-*codes* amb seguretat forta. Si no diem el contrari sempre suposarem que els AMD-*codes* amb els que tractem són amb seguretat forta.

En els (S, G, δ) AMD-*codes* amb seguretat dèbil demanarem que per a qualsevol $\Delta \in \mathcal{G}$ i per a qualsevol $s \in S$ triat de manera uniformement aleatòria es compleixi:

$$\Pr[\mathcal{D}(\mathcal{E}(s) + \Delta) \notin \{s, \perp\}] \leq \delta$$

En els (S, G, δ) AMD-*codes* amb seguretat forta la condició es compleix per a qualsevol $s \in \mathcal{S}$ i qualsevol $\Delta \in \mathcal{G}$. És a dir que s podria haver estat triat de manera no aleatòria (per exemple un adversari la podria haver triat maliciosament).

Un AMD-*code* s'anomena sistemàtic si $\mathcal{S}$ és un grup i la codificació és duu a terme de la següent manera:

$$\begin{aligned} \mathcal{E} : \mathcal{S} &\to \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \\ s &\mapsto (s, x, f(x, s)) \end{aligned}$$

per a alguna funció f i $x \in_R \mathcal{G}_1$. La descodificació ve donada per:

$$\mathcal{D}(\tilde{s}, \tilde{x}, \tilde{\sigma}) = \begin{cases} \tilde{s} & \text{si } \tilde{\sigma} = f(\tilde{x}, \tilde{s}) \\ \perp & \text{altrament} \end{cases}$$

Definirem també una família d'AMD-*codes* com un conjunt d'aquests tals que, per a qualssevol $\kappa, u \in \mathbb{N}$ existeix un (S, G, δ) AMD-*code* de la família amb $S \geq 2^u$ i $\delta \leq 2^{-\kappa}$. D'aquesta definició cal especificar que el grup $\mathcal{G}$ pot ser diferent en cada AMD-*code* de la família.

El que voldrem en un AMD-*code* és que $\mathcal{G}$ no sigui massa més gran que $\mathcal{S}$ és a dir, que la redundància que haurem d'afegir a la informació sigui la mínima possible. Per aquest motiu definim la *tag size* d'un (S, G, δ) AMD-*code* com

$$\varpi = \log(G) - \log(S)$$

També definirem un concepte més general aplicable a les famílies d'AMD-*codes*, la *tag size* efectiva $\varpi^*(\kappa, u)$ en referència als valors $\kappa, u \in \mathbb{N}$ d'una família d'AMD-*codes* com

$$\varpi^*(\kappa, u) = \min(\log(G)) - u$$

on el mínim es pren sobre tots els (S, G, δ) AMD-*codes* dins de la família amb $S \geq 2^u$ i $\delta \leq 2^{-\kappa}$.

5.3 Fites

En aquesta secció exposarem un seguit de fites sobre les AMD-*codes*.

Teorema 5.1. *Un* (S, G, δ) *AMD-*code *compleix que:*

$$G \geq \frac{S-1}{\delta} + 1 \quad \textit{en el cas de seguretat dèbil}$$

$$G \geq \frac{S-1}{\delta^2} + 1 \quad \textit{en el cas de seguretat forta}$$

Demostració. Siguin $(\mathcal{E}, \mathcal{D})$ les funcions de codificar i descodificar respectivament del nostre AMD-*code*.

$$\mathcal{E} : \mathcal{S} \rightarrow \mathcal{G} \quad \text{i} \quad \mathcal{D} : \mathcal{G} \rightarrow \mathcal{S} \times \{\perp\}$$

Per a qualsevol $s \in \mathcal{S}$ considerem $\mathcal{D}^{-1}(s) = \{e \in \mathcal{G} \ : \ \mathcal{D}(e) = s\}$. Clarament $\mathcal{D}^{-1}(s) \cap \mathcal{D}^{-1}(s') = \emptyset$ per a qualsevol $s' \neq s$ i $|\mathcal{D}^{-1}(s)| \geq 1$.

Considerem el cas amb seguretat dèbil. Sigui $s \in \mathcal{S}$ i prenem $\Delta \neq 0$ aleatòriament de $\mathcal{G}$ i independentment de s. La probabilitat que el succés $\mathcal{D}(\mathcal{E}(s) + \Delta) \neq \{s, \perp\}$ és menor o igual que δ, per tant:

$$\delta \geq \Pr[\mathcal{E}(s) + \Delta \in \cup_{s \neq s'} \mathcal{D}^{-1}(s')] = \frac{|\cup_{s \neq s'} \mathcal{D}^{-1}(s')|}{G-1} \geq \frac{S-1}{G-1}$$

on en la primera desigualtat es considera Δ fix i en la primera igualtat es considera s fix i es té en compte que si la desigualtat val per a qualsevol valor fixat, també serà certa per a un valor aleatori.

Considerem ara el cas amb seguretat forta. Per a qualsevol $s \in \mathcal{S}$ tenim que $\delta \geq 1/|\mathcal{D}^{-1}(s)|$ perquè un adversari sempre coneix $\mathcal{D}^{-1}(s)$ (ja que coneix s). Per tant, sempre pot provar canviant s per un valor d'aquest conjunt. Això implica $|\mathcal{D}^{-1}(s)| \geq 1/\delta$ i , per tant

$$\delta \geq \Pr[\mathcal{E}(s) + \Delta \in \cup_{s \neq s'} \mathcal{D}^{-1}(s')] = \frac{|\cup_{s \neq s'} \mathcal{D}^{-1}(s')|/\delta}{G-1} \geq \frac{S-1}{G-1}$$

□

Amb aquest teorema demostrat podem trobar fàcilment fites per a les *tag sizes* efectives que ens diran que si volem AMD-*codes* amb seguretat $2^{-\kappa}$, dèbil o forta és inevitable que el missatge augmenti amb κ o 2κ respectivament.

Corol·lari 5.2. *Totes les famílies d'AMD-codes tenen una* tag size *efectiva fitada per:*

$$\varpi^*(\kappa, u) \geq \kappa - 2^{-u+1} \quad \textit{en el cas de seguretat dèbil}$$

$$\varpi^*(\kappa, u) \geq 2\kappa - 2^{-u+1} \quad \textit{en el cas de seguretat forta}$$

Demostració. Considerem el cas amb seguretat dèbil. Sigui un (S, G, δ) AMD-*code* amb $S \geq 2^u$ i $\delta \leq 2^{-\kappa}$.

$$\log(G) - u \geq \log(G) - \log(S) \geq \log\left(\frac{G-1}{S-1}\frac{S-1}{S}\right) = \log\left(\frac{G-1}{S-1}\right) + \log\left(1 - \frac{1}{S}\right) \geq \kappa - \frac{2}{S}$$

on l'última desigualtat prové del Teorema 5.1 i de la fita $\log(1-x) \geq -2x$ per a $0 \leq x \leq 1/2$.

En el cas amb seguretat forta procedim de la mateixa manera canviant l'última desigualtat

$$\log(G) - u \geq \log\left(\frac{G-1}{S-1}\right) + \log\left(1 - \frac{1}{S}\right) \geq 2\kappa - \frac{2}{S}$$

□

5.4 Relació amb la combinatòria

Com veurem en aquesta secció, les estructures que formen certs AMD-*codes* són estructures combinatòries. Recíprocament, podrem construir AMD-*codes* amb seguretat dèbil i també forta a partir d'aquestes estructures. Exposem en primer lloc la construcció amb seguretat dèbil fent primer una definició prèvia.

Sigui $\mathcal{G}$ un grup finit d'ordre G, diem que un subconjunt $V \subset \mathcal{G}$ de mida S és un (S, G, t) conjunt de diferències fitat si la llista de diferències $v_i - v_j$ on $v_i, v_j \in V$, conté cada element no nul de $\mathcal{G}$ com a molt t vegades. Notem que la definició estàndard en combinatòria de conjunt de diferències conté cada element exactament t vegades.

Direm que un AMD-*code* és determinista si la seva funció de codificar $\mathcal{E}$ també ho és. Amb aquestes definicions podem exposar el següent resultat:

Teorema 5.3. *Si $V \subset \mathcal{G}$ és un (S, G, t) conjunt de diferències fitat llavors l'AMD-*code *següent:*

$$\mathcal{E}: \begin{array}{ccc} V & \to & \mathcal{G} \\ s & \mapsto & s \end{array} \quad i \quad \mathcal{D}(s) = \begin{cases} s & si \ \ s \in V \\ \perp & altrament \end{cases}$$

és un (S, G, δ) *AMD-*code *determinista amb seguretat dèbil i amb* $\delta = t/S$*. I recíprocament, per a un* (S, G, δ) *AMD-*code *determinista qualsevol amb funcions per codificar i descodificar* $(\mathcal{E}, \mathcal{D})$*, el conjunt* $V = \mathcal{E}(\mathcal{S}) = \{\mathcal{E}(s) \ : \ s \in \mathcal{S}\} \subset \mathcal{G}$ *és un* (S, G, t) *conjunt de diferències fitat amb* $t = \delta S$*.*

Demostració. Està clar que $(\mathcal{E}, \mathcal{D})$ és un AMD-*code* amb seguretat dèbil. Només ens queda argumentar el valor δ. Per les propietats de V, per a cada element no nul $\Delta \in \mathcal{G}$ existeixen com a molt t elements $s \in V$ tals que $s + \Delta \in V$, per tant si prenem $s \in V$ de manera uniforme i aleatòria i Δ independentment de s, el succés $s + \Delta \in V$ (equivalent al fet que l'adversari guanyi) passarà amb probabilitat t/S. La implicació contrària es raona de manera similar. □

Per a la construcció d'AMD-*codes* amb seguretat forta ens caldrà una altra definició relacionada amb la combinatòria. Sigui $\mathcal{G}$ un grup finit d'ordre G, $\mathcal{S}$ un conjunt finit de cardinal S, per simplicitat escriurem $\mathcal{S} = \{1, \ldots, S\}$ i siguin $V_1, \ldots, V_S$ conjunts disjunts i no buits de $\mathcal{G}$. Direm que $(\mathcal{G}, V_1, \ldots, V_S)$ és una estructura diferenciada.

Direm que un AMD-*code* té selecció uniforme si per a cada $s \in \mathcal{S}$, la codificació $\mathcal{E}(s)$ està distribuïda uniformement en $\mathcal{D}^{-1}(s) = \{e \in \mathcal{G} \ : \ \mathcal{D}(e) = s\}$. Finalment definirem els següents paràmetres:

$$t_i = \max_{\Delta \in \mathcal{G}} |(V_i + \Delta) \cap \bigcup_{i \neq j} V_j|$$

Teorema 5.4. *Si tenim una estructura diferenciada* $(\mathcal{G}, V_1, \ldots, V_S)$ *i els paràmetres definits anteriorment* $t_1, \ldots, t_S$*, el següent AMD-code:*

$$\begin{array}{rccc} \mathcal{E}: & \{1, \ldots, S\} & \to & \mathcal{G} \\ & s & \mapsto & \tilde{s} \end{array} \quad i \quad \mathcal{D}(s) = \begin{cases} s & si \ \ \exists s : \tilde{s} \in V_s \\ \perp & altrament \end{cases}$$

on $\tilde{s}$ *es tria de manera uniforme i aleatòria entre els valors de* V_s*, és un* (S, G, δ) *AMD-*code *amb seguretat forta i selecció uniforme on* $\delta = \max_i t_i/|V_i|$*. I recíprocament, per a qualsevol* (S, G, δ) *AMD-code amb seguretat forta i selecció uniforme, els conjunts* $V_s = \mathcal{D}^{-1}(s)$ *per a* $s \in \mathcal{S}$ *formen una estructura diferenciada amb* $t_s \leq \delta |V_s|$*.*

Demostració. Fixem un valor s arbitrari, sigui $\tilde{s}$ la seva codificació uniformement distribuïda en V_s i sigui Δ la diferència afegida a $\tilde{s}$ per part de l'adversari, independent de $\tilde{s}$. Llavors $\tilde{s} + \Delta$ està uniformement distribuït dins de $V_s + \Delta$ i la probabilitat que caigui dins d'algun V_j $(j \neq s)$ és com a molt t_s/V_s. L'altra implicació s'argumenta de manera similar. □

Lema 5.5. *Si l'AMD-*code *del Teorema 5.4 és sistemàtic, l'estructura diferenciada compleix que*

$$t_i = \max_{\substack{\Delta \in \mathcal{G} \\ j \neq i}} |(V_i + \Delta) \cap V_j|$$

Demostració. Si suposem que l'AMD-*code* és sistemàtic sabem que per a qualsevol $s \in \mathcal{S}$ tenim que $\mathcal{E}(s) = (s, x, f(s,x))$ amb $x \in_R \mathcal{G}_1$ (recordem la definició de la Secció 5.2). Per tant $V_i = \{(i, x_1, f(i,x_1)), (i, x_2, f(i,x_2)), \cdots, (i, x_n, f(i,x_n))\}$ (on $n = |\mathcal{G}_1|$).
Per altra banda

$$t_i = \max_{\Delta \in \mathcal{G}} |(V_i + \Delta) \cap \bigcup_{i \neq j} V_j| = \max_{\Delta \in \mathcal{G}} |\bigcup_{i \neq j} (V_i + \Delta) \cap V_j|$$

Però $V_i + \Delta = \{(i + \Delta_i, x_1 + \Delta_x, f(i,x_1) + \Delta_f), (i + \Delta_i, x_2 + \Delta_x, f(i,x_2) + \Delta_f), \cdots, (i + \Delta_i, x_n + \Delta_x, f(i,x_n) + \Delta_f)\}$ i aquest conjunt només pot tenir intersecció no nul·la amb $V_{i+\Delta_i}$ és a dir, com a mínim, tots els V_j amb $j \neq i$ excepte un tindran intersecció nul·la amb $V_i + \Delta$. Per tant,

$$t_i = \max_{\Delta \in \mathcal{G}} |\bigcup_{i \neq j} (V_i + \Delta) \cap V_j| = \max_{\substack{\Delta \in \mathcal{G} \\ j \neq i}} |(V_i + \Delta) \cap V_j|$$

□

5.5 Aplicacions

En aquesta secció mostrarem dues aplicacions dels AMD-*codes* per construir AMD-*codes* amb seguretat forta i KMS-MACs. Primer de tot exposarem un AMD-*code* amb seguretat forta construït a partir d'un AMD-*code* amb seguretat dèbil i un MAC, és a dir un codi d'autentificació de missatges. Considerem el MAC següent:

$$\begin{aligned} A: \mathcal{S} \times \mathcal{K} &\to \mathcal{T} \\ (s,k) &\mapsto \sigma \end{aligned}$$

on k és la clau privada i considerem p_S la probabilitat que té l'adversari de guanyar en un atac per substitució, és a dir,

$$p_S = \max_{s \neq \tilde{s}} \Pr[A(\tilde{s}, k) = \tilde{\sigma} | A(s,k) = \sigma]$$

Considerem per últim aquest (S', G', δ') AMD-*code*:

$$\mathcal{E}' : \mathcal{S}' \to \mathcal{G}'$$

amb seguretat dèbil i amb $\mathcal{S}' = \mathcal{K}$.

Teorema 5.6. *El següent AMD-*code*:*

$$\begin{aligned} \mathcal{E}: \mathcal{S} &\to \mathcal{S} \times \mathcal{G}' \times \mathcal{T} \\ s &\mapsto (s, \mathcal{E}'(k), A(s,k)) \end{aligned}$$

$$\mathcal{D}(s, e', \sigma) = \begin{cases} s & si \;\; \mathcal{D}'(e') \neq \perp \; i \; \sigma = A(s, \mathcal{D}'(e')) \\ \perp & altrament \end{cases}$$

*on k es pren aleatòriament de $\mathcal{K}$, és un $(S, S{\cdot}G'{\cdot}T, \delta)$ AMD-*code *amb seguretat forta on $\delta = \delta' + p_S$. Si el codi subjacent $\mathcal{E}'$ és sistemàtic llavors $\delta = \max\{\delta, p_S\}$.*

Demostració. Evidentment les mides de l'AMD-*code* concorden. Només resta comprovar el valor de δ. Fixem un $s \in \mathcal{S}$ arbitrari i una translació $\Delta = (\Delta_s, \Delta_{e'}, \Delta_\sigma) \in \mathcal{S} \times \mathcal{G}' \times \mathcal{T}$ també arbitrària amb $\Delta_s \neq 0$ Sigui $e = (s, e', \sigma) = \mathcal{E}(s) = (s, \mathcal{E}'(k), A(s, k))$ per a una k aleatòria. Volem saber la probabilitat que $\mathcal{D}(e + \Delta) = s + \Delta_s$ per tant ens fixarem en les condicions que ens demanen per a la descodificació.
Per les propietats d'$\mathcal{E}'$, la probabilitat que $\mathcal{D}'(e' + \Delta_{e'}) \neq \{\perp, k\}$ és com a molt δ'. Per altra banda, si passés que $\mathcal{D}'(e' + \Delta_{e'}) = k$, llavors la probabilitat que $\sigma + \Delta_\sigma = A(s + \Delta_s, k)$ és com a molt p_S. Per tant la probabilitat que $\mathcal{D}(e + \Delta) = s + \Delta_s$ és com a molt $\delta + p_S$.

En el cas que el codi $\mathcal{E}'$ sigui sistemàtic, la codificació e' té com a primera component k i per tant, $\Delta_{e'} = (\Delta_k, \Delta_{e'-k})$. Això ens permet fer una distinció per casos segons si $\Delta_k \neq 0$ o $\Delta_k = 0$. Si $\Delta_k \neq 0$ llavors $\mathcal{D}'(e') \neq \perp$ amb probabilitat com a molt δ' (de fet les úniques descodificacions possibles per a $\mathcal{D}'(e')$ són $\{\perp, k + \Delta_k\}$ ja que el codi és sistemàtic). Si $\Delta_k = 0$, com que estem considerant atacs intel·ligents (és a dir, la nostra finalitat és modificar k), $\Delta_{e'-k} = 0$ i això implica que $\mathcal{D}'(e') = k$. Llavors la probabilitat que $\sigma + \Delta_\sigma = A(s + \Delta_s, k)$ és com a molt p_S. La probabilitat global és doncs $\delta = \max\{\delta', p_S\}$. □

A continuació establirem una fita per a la *tag size* efectiva dels AMD-*codes* amb seguretat forta obtinguts a partir d'aquesta construcció. Tanmateix, abans exposarem un lema previ que no demostrarem i que necessitarem per a la demostració posterior de la fita.

Lema 5.7. *Qualsevol codi d'autentificació de missatges sistemàtic*

$$A : \mathcal{S} \times \mathcal{K} \to \mathcal{T}$$

amb probabilitat d'èxit p_S en un atac per substitució ha de complir

$$|\mathcal{T}| \geq 1/p_S \quad i \quad |\mathcal{K}| \geq 1/p_S^2$$

Corol·lari 5.8. *Per a qualsevol AMD-*code *amb seguretat forta obtingut amb la construcció del Teorema 5.6 tenim que la seva* tag size *efectiva compleix*

$$\varpi^*(\kappa, u) \geq 4\kappa - 2^{-2\kappa+1}$$

Demostració. Si volem aconseguir una probabilitat d'error $\delta \leq 2^{-\kappa}$, això implica $\delta' \leq 2^{-\kappa}$ i $p_S \leq 2^{-\kappa}$. Pel Lema 5.7 sabem que $\log(|\mathcal{K}|) \geq 2\kappa$ i $\log(|\mathcal{T}|) \geq \kappa$.

Fixem-nos ara en el codi subjacent a la construcció. Sabem que $\mathcal{S}' = \mathcal{K}$ i pel Corol·lari 5.2 tenim que

$$\varpi^*_{\mathcal{E}'}(\kappa, 2\kappa) = \min(\log(|\mathcal{G}'|)) - \log(|\mathcal{K}|) \geq \kappa - 2^{-2\kappa+1}$$

per tant $\log(|\mathcal{G}'|) \geq \log(|\mathcal{K}|) + \kappa - 2^{-2\kappa+1} \geq 3\kappa - 2^{-2\kappa+1}$.

Finalment, la *tag size* efectiva del codi que volem serà

$$\begin{aligned} \varpi^*(\kappa, u) &= \min[\log(|\mathcal{S}|) + \log(|\mathcal{G}'|) + \log(|\mathcal{T}|)] - \log(|\mathcal{S}|) = \\ &= \min(\log(|\mathcal{G}'|) + \log(|\mathcal{T}|)) \geq \\ &\geq 3\kappa - 2^{-2\kappa+1} + \kappa = 4\kappa - 2^{-2\kappa+1} \end{aligned}$$

□

Vista aquesta aplicació, exposarem ara com fer servir els AMD-*codes* per construir KMS-MACs és a dir, codis d'autentificació de missatges amb seguretat contra manipulacions de la clau.

Un (S, G, T, δ) KMS-MAC és una funció $A : \mathcal{S} \times \mathcal{G} \to \mathcal{T}$ on $S = |\mathcal{S}|$, $T = |\mathcal{T}|$ i $\mathcal{G}$ és un grup d'ordre G tal que per a qualssevol $s \neq \tilde{s} \in \mathcal{S}$, $\sigma, \tilde{\sigma} \in \mathcal{T}$ i qualsevol $\Delta \in \mathcal{G}$ tenim que

$$\Pr[A(\tilde{s}, k + \Delta) = \tilde{\sigma} | A(s, k) = \sigma] \leq \delta$$

on la clau k és triada de manera uniforme i aleatòria.

Tal i com hem fet amb els AMD-*codes* treballarem amb famílies de KMS-MAC's. Això ens permetrà obtenir un resultat anàleg al Lema 5.7. Així doncs, per a qualsevol KMS-MAC amb $\delta \leq 2^{-\kappa}$ tenim que la mida de $\mathcal{G}$ i la mida de $\mathcal{T}$ compleixen:

$$\log(G) \geq 2\kappa \quad \text{i} \quad \log(T) \geq \kappa$$

Ara construirem un KMS-MAC a partir d'un AMD-*code* sistemàtic.

Teorema 5.9. *Sigui*

$$\begin{aligned} \mathcal{E} : \mathcal{S} &\to \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \\ s &\mapsto (s, x, f(x, s)) \end{aligned}$$

un $(S, S \cdot G_1 \cdot G_2, \delta)$ *AMD-*code *sistemàtic. Llavors la funció:*

$$\begin{aligned} A : \mathcal{S} \times (\mathcal{G}_1 \times \mathcal{G}_2) &\to \mathcal{G}_2 \\ (s, (x_1, x_2)) &\mapsto f(x_1, s) + x_2 \end{aligned}$$

és un $(S, G_1 \cdot G_2, G_2, \delta)$ *KMS-MAC.*

Demostració. Prenem $K = (x_1, x_2) \in \mathcal{G}_1 \times \mathcal{G}_2$ triat de manera uniforme i aleatòria i considerem $\Delta = (\Delta_1, \Delta_2) \in \mathcal{G}_1 \times \mathcal{G}_2$; $\sigma, \sigma' \in \mathcal{G}_2$ i $s', s' \in \mathcal{S}$ on $s \neq s'$.

El succés $MAC(s, K) = \sigma$ vol dir $\sigma = f(x_1, s) + x_2$ que és el mateix que $x_2 = -f(x_1, s) + \sigma$ i l'anomenem succés E_1. El succés $MAC(s', K + \Delta) = \sigma'$ vol dir $\sigma' = f(x_1 + \Delta_1, s') + (x_2 + \Delta_2)$ que és el mateix que $f(x_1 + \Delta_1, s') = -x_2 + \sigma' - \Delta_2$ i l'anomenem succés E_2. Considerarem també un succés auxiliar E_2' que serà $f(x_1 + \Delta_1, s') = f(x_1, s) + \Delta_f$ on $\Delta_f = -\sigma + \sigma' - \Delta_2$.

Volem trobar $\Pr[E_2|E_1]$. En primer lloc demostrarem que $\Pr[E_2|E_1] = \Pr[E_2'|E_1]$ ja que si suposem E_1, és a dir $x_2 = -f(x_1, s) + \sigma$ llavors

$$-x_2 + \sigma' - \Delta_2 = -(-f(x_1, s) + \sigma) + \sigma' - \Delta_2 = f(x_1, s) + (-\sigma + \sigma' - \Delta_2) = f(x_1, s) + \Delta_f$$

Finalment ens fixarem que E_2' i E_1 són independents perquè E_2' no depèn de x_2, i aquest valor x_2 es tria aleatòriament dins de $\mathcal{G}_2$. Per tant el fet que x_2 sigui o no igual a $-f(x_1, s) + \sigma$ no afectarà altres esdeveniments que no involucrin x_2. Així doncs $\Pr[E_2'|E_1] = \Pr[E_2']$ i tenim que

$$\Pr[\text{MAC}(s', K + \Delta) = \sigma' \mid \text{MAC}(s, K) = \sigma] = \Pr[f(x_1 + \Delta_1, s') = f(x_1, s) + \Delta_f] \leq \delta$$

on l'última desigualtat és deguda a la seguretat de l'AMD-*code* ja que $s \neq s'$. □

5.6 Primeres construccions

En aquesta secció plantejarem algunes de les primeres construccions pràctiques d'AMD-*codes* que es van plantejar.

Ogata i Kurosawa [9] van plantejar una construcció d'un esquema de compartició de secrets robusta basada en el següent AMD-*code*. Sigui q un primer de manera que $p = q^2 + q + 1$ també és un primer. Sigui $B \subset \{0, \ldots, p-1\}$ un conjunt de diferències pla, és a dir, un conjunt tal que les $q(q+1) = p - 1$ diferències formades per totes les parelles possibles d'elements de B són exactament els nombres $1, \ldots, p-1$. Està demostrat que aquests conjunts de diferències existeixen. Llavors l'aplicació:

$$\begin{aligned} E: B &\to \mathbb{Z}_p \\ s &\mapsto s \end{aligned}$$

és un $(q+1, q^2+q+1, 1/(q+1))$ AMD-*code* amb seguretat dèbil. A més a més la *tag size* d'aquest codi és $\varpi = \log(q^2+q+1) - \log(q+1)$ que està entre els valors $\log(q)$ i $\log(q+1)$.

Cabello, Padró i Sáez [1] van proposar dues construccions més d'AMD-*codes*. Una amb seguretat dèbil i l'altra amb seguretat forta. Per a qualsevol cos finit $\mathbb{F}$ d'ordre q la construcció:

$$\begin{aligned} E: \mathbb{F} &\rightarrow \mathbb{F} \times \mathbb{F} \\ s &\mapsto (s, s^2) \end{aligned}$$

és un $(q, q^2, 1/q)$ AMD-*code* amb seguretat dèbil i

$$\begin{aligned} E: \mathbb{F} &\rightarrow \mathbb{F} \times \mathbb{F} \times \mathbb{F} \\ s &\mapsto (s, x, sx) \end{aligned}$$

és un $(q, q^3, 1/q)$ AMD-*code* amb seguretat forta.

Aquestes dues construccions es podrien considerar òptimes ja que si ens fixem en les fites del Lema 5.1 per a codis amb seguretat dèbil, l'òptim s'assoleix quan:

$$G = \frac{S-1}{\delta} + 1 \rightarrow G \approx \frac{S}{\delta}$$

Per tant:

$$\log(G) \approx \log(S) - \log(\delta) \rightarrow \varpi = \log(G) - \log(S) \approx -\log(\delta)$$

Per a codis amb seguretat forta, seguint el mateix procediment arribem al resultat següent:

$$\varpi = \log(G) - \log(S) \approx -2\log(\delta)$$

Aquest fet es compleix en les dues construccions proposades anteriorment. El principal problema d'aquestes construccions és que no es poden escalar, és a dir, no es pot augmentar tant com vulguem la mida del conjunt de sortida perquè tant la probabilitat d'error δ com la mida del conjunt d'arribada G depenen d'aquest.

De fet, en termes de la *tag size* efectiva tenim que

$$\varpi^*(\kappa, u) \geq 2u - u = u \quad \text{en el cas de seguretat dèbil}$$

$$\varpi^*(\kappa, u) \geq 3u - u = 2u \quad \text{en el cas de seguretat forta}$$

Aquests dos valors estan molt allunyats de l'òptim que s'obtindria amb les fites del Corol·lari 5.2 per a les *tag sizes* efectives si u és considerablement més gran que κ.

En el mateix paper [1] es presenta una construcció d'un AMD-*code* amb seguretat dèbil que sí que és escalable és a dir, amb una redundància de llargada constant podem augmentar la llargada del missatge tant com vulguem sense comprometre la seguretat del codi.

Per a aquesta construcció considerarem el missatge $s \in \mathbb{F}_{q^r}$ i veurem aquest cos com un subespai de dimensió r sobre $\mathbb{F}_q$. Triarem també un epimorfisme ϕ tal que:

$$\begin{aligned} \phi: \mathbb{F}_{q^r} &\rightarrow \mathbb{F}_q \\ s &\mapsto \phi(s) \end{aligned}$$

Amb aquestes definicions fetes

$$\begin{aligned} E: \mathbb{F}_{q^r} &\rightarrow \mathbb{F}_{q^r} \times \mathbb{F}_q \\ s &\mapsto (s, \phi(s^2)) \end{aligned}$$

és un $(q^r, q^{r+1}, 1/q)$ AMD-*code* amb seguretat dèbil, amb $\varpi = log(q)$ i una *tag size* efectiva molt propera a l'òptim ja que $\varpi^*(\kappa, u) \approx \kappa$. A més a més podem augmentar la llargada del missatge s (és a dir augmentar r) mantenint constant la probabilitat d'error $1/q$.

Més endavant veurem que en el cas dels AMD-*codes* amb seguretat forta (o simplement AMD-*codes*) no serà possible aconseguir resultats tan bons com aquests.

5.7 Construccions recents

En aquesta secció revisarem les construccions més recents d'AMD-*codes* que incorporen codis correctors d'errors que ajuden a aconseguir AMD-*codes* amb seguretat forta molt propers a l'òptim.

Vegem-ne primer una visió general per passar després a les construccions específiques. Sigui $\mathbb{F}$ un cos finit de mida q i sigui $\mathcal{C} \subset \mathbb{F}^n$ un codi corrector d'errors de dimensió k. En el cas que ens ocupa (i contrastant amb la tendència general quan s'estudien codis correctors d'errors) voldrem n molt gran.

Sigui $C : \mathbb{F}^k \rightarrow \mathcal{C}$ una funció codificadora i bijectiva. Considerem el següent AMD-*code*:

$$\begin{aligned} \mathcal{E}: \mathbb{F}^k &\rightarrow \mathbb{F}^k \times \mathbb{Z}_n \times \mathbb{F} \\ s &\mapsto (s, x, [C(s)]_x) \end{aligned}$$

on $x \in_R \mathbb{Z}_n$ i $[C(s)]_x$ és la x-èssima coordenada del vector $C(s) \in \mathbb{F}^n$.

Veiem doncs que aquest AMD-*code* tindrà una probabilitat d'error baixa si les paraules del codi contenen poques vegades el mateix símbol (fins i tot si fem una rotació cíclica de les paraules del codi).

Formalment, per a qualsevol $v = (v_0, \ldots, v_{n-1}) \in \mathbb{F}^n$ sigui $M(v)$ el màxim nombre de vegades que apareix qualsevol símbol a la paraula del codi v, és a dir,

$$M(v) = \max_{a \in \mathbb{F}} |\{i : v_i = a\}|$$

Considerarem que dues paraules del codi són properes si $M(c - c')$ és gran. I definirem

$$\mu(\mathcal{C}) = \max_{c \neq c' \in \mathcal{C}} M(c - c')$$

que ens mesurarà com n'estan de properes dues paraules qualssevol del codi.

Finalment definirem la clausura de $\mathcal{C} \subseteq \mathbb{F}^n$ com:

$$\mathrm{cl}(\mathcal{C}) = \bigcup_{t \in \mathbb{Z}_n} \mathrm{rot}_t(\mathcal{C})$$

on aquesta rotació s'ha d'entendre com una translació de t posicions mòdul n, és a dir $\mathrm{rot}_t(c) = (c_{0+t}, \ldots, c_{n-1+t})$ on $c = (c_0, \ldots, c_{n-1})$. Amb aquestes definicions podem enunciar el teorema següent:

Teorema 5.10. *El codi $\mathcal{E}$ és un (q^k, nq^{k+1}, δ) AMD-*code *amb $\delta = \mu(cl(\mathcal{C}))/n$ sempre que $cl(\mathcal{C})$ sigui la unió disjunta de les $rot_t(\mathcal{C})$ (és a dir $rot_t(\mathcal{C}) \cap \mathcal{C} = \emptyset\ \forall t \in \mathbb{Z}_n^*$) i $\delta = 1$ altrament.*

Demostració. És clar que les dimensions són correctes i que $\mathcal{E}$ és un AMD-*code* així doncs, només falta argumentar el valor de δ. Fixem-nos que si existeix t tal que $\mathrm{rot}_t(\mathcal{C}) \cap \mathcal{C} \neq \emptyset$ llavors prenent aquestes dues paraules del codi podem dur a terme l'estratègia següent: prenem aquestes dues paraules del codi c i $\sigma_t(c)$, busquem les seves antiimatges, s i s' respectivament. A continuació denotem $\Delta_s = s' - s$ i enviem $(s + \Delta_s, x + t, [C(s)]_x) = (s, x, [C(s)]_x) + (\Delta_s, t, 0)$. En aquest punt l'adversari ha aconseguit canviar l'entrada s sense que es pugui detectar ja que:

$$[C(s)]_x = [C(s')]_{x+t} = [C(s + s' - s)]_{x+t} = [C(s + \Delta_s)]_{x+t}$$

Si la condició $\mathrm{rot}_t(\mathcal{C}) \cap \mathcal{C} = \emptyset\ \forall t \in \mathbb{Z}_n^*$ es compleix, expressarem l'AMD-*code* com en el Teorema 5.4. Per a cada $s \in \mathbb{F}^k$ definirem el conjunt $V_s = \{(s, x, e)\ :\ e = [C(s)]_x\}$. De fet estem considerant un cas concret de la construcció general feta al Teorema 5.4. En el nostre cas $|V_s| = n$, així doncs només falta determinar els valors t_s. Fixem un valor $s \in \mathbb{F}^k$ i una translació arbitrària $(\Delta_s, \Delta_x, \Delta_e) \in \mathbb{F}^k \times \mathbb{Z}_n \times \mathbb{F}$ amb $\Delta_s \neq 0$. Hem de comptar el nombre d'elements de V_s que cauen dins $V_{s+\Delta_s}$ quan apliquem la translació $(\Delta_s, \Delta_x, \Delta_e)$, és a dir el nombre d'elements $x \in \mathbb{Z}_n$ tals que

$$[C(s)]_x + \Delta_e = [C(s + \Delta_s)]_{x+\Delta_x} = [\mathrm{rot}_{\Delta_x}(C(s + \Delta_s))]_x$$

Escrivint $c = C(s)$ i $c' = C(s + \Delta_s)$, aquest nombre que busquem és com a molt $M(c - \mathrm{rot}_{\Delta_x}(c'))$. Aquest valor, a la vegada està fitat superiorment per $\mu(\mathrm{cl}(\mathcal{C}))$ (estem suposant $c \neq \mathrm{rot}_{\Delta_x}(c')$). Amb tot això el Teorema 5.4 combinat amb el Lema 5.5 implica $\delta = \mu(\mathrm{cl}(\mathcal{C}))/n$. □

Si volem aconseguir bons AMD-*codes* amb aquesta construcció hem d'aconseguir que el codi subjacent amb totes les seves rotacions (és a dir $cl(\mathcal{C})$) tingui les paraules prou llunyanes (en el sentit que $M(v)$ sigui petit).

Això ens porta a l'estudi dels codis cíclics que són els que finalment haurem de tractar (perquè de fet cl($\mathcal{C}$) sempre és un codi cíclic). L'estudi doncs, es basarà en la $\mu(\mathcal{C})$ dels codis cíclics per veure fins on podem arribar.

Els problemes que presenta aquest estudi és que per als codis correctors d'errors sempre es treballa simplement amb la distància de Hamming, que compta el nombre de posicions amb símbols diferents (o nombre de zeros en el cas dels codis lineals) mentre que, en aquest cas, hem de treballar amb M que estudia el màxim nombre de símbols iguals que pot arribar a tenir una paraula del codi.

És fàcil deduir que $\mu(\mathcal{C}) \geq n - d$ on d és la distància de Hamming, ja que la μ compta el nombre símbols iguals que pot tenir qualsevol diferència de paraules del codi $c - c'$. D'altra banda la distància de Hamming compta el nombre de símbols diferents pot tenir $c - c'$ (és a dir que $n - d$ compta el nombre de zeros que pot arribar a tenir $c - c'$). Per tant la μ, en particular també haurà de tenir en compte aquests zeros que es compten amb la distància de Hamming.

Fent servir la fita de Singleton podem arribar al següent resultat:

$$\mu(\mathcal{C}) \geq n - d \geq k - 1$$

En els codis no lineals s'hauria de substituir k per $\log_q(|\mathcal{C}|)$. I això ens dóna una fita per als codis obtinguts amb la construcció anterior:

$$\delta = \frac{\mu(\mathcal{C})}{n} \geq \frac{n-d}{n} \geq \frac{k-1}{n}$$

fent la mateixa substitució en el cas no lineal. Amb aquest resultat ens podem adonar que amb aquesta construcció no podrem arribar mai al valor ideal (que seria $\delta = 1/n$) és a dir, que el numerador mai podrà ser 1. Però aquesta construcció val per a qualsevol AMD-*code* sistemàtic ja que si considerem:

$$\begin{aligned} \mathcal{E} : \mathcal{S} &\rightarrow \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \\ s &\mapsto (s, x, f(x, s)) \end{aligned}$$

un AMD-*code* sistemàtic genèric amb $|\mathcal{G}_1| = n$, llavors $\mathcal{C} = \{c \in \mathcal{G}_2^n \ / \ c = C(s) = (f(x_1, s), f(x_2, s), \ldots, f(x_n, s))\}$ és un codi corrector d'errors.

Per tant podem concloure que, amb qualsevol AMD-*code* sistemàtic no podrem aconseguir una probabilitat d'error menor que $(k-1)/n$, i això confirma el fet que aquest tipus de codis no es poden escalar del tot (és a dir fins al valor ideal). També es creu que amb els AMD-*codes* no sistemàtics no es podrien obtenir resultats millors que els aconseguits amb els AMD-*codes* sistemàtics tot i que no s'ha demostrat encara.

A continuació exposarem una construcció insegura d'un AMD-*code* basada en polinomis i proposada en [8] com a segura (amb seguretat forta). La construcció és la següent: considerem $\mathbb{F}$ un cos de q elements, llavors la funció per codificar és

$$\begin{aligned} \mathcal{E}: \mathbb{F}^d &\to \mathbb{F}^d \times \mathbb{F} \times \mathbb{F} \\ s &\mapsto (s, x, f(x,s)) \end{aligned}$$

on $x \in_R \mathbb{F}$ i $f(x,s) = s_1 x + \cdots + s_d x^d$.

Aquesta construcció va ser proposada com a segura però veurem que es pot trencar la seguretat perquè l'estratègia permetrà a l'adversari canviar el missatge sense ser descobert. Fixem-nos que

$$f(x+\Delta_x, s) = \sum_{i=1}^{d} s_i (x+\Delta_x)^i = \sum_{i=1}^{d} s'_i x^i + \Delta_f = f(x, s') + \Delta_f$$

per a algun s' i algun $\Delta_f \in \mathbb{F}$ on aquests dos valors són calculables de manera eficient donats s i Δ_x. Recordem que si suposem seguretat forta, l'adversari té accés a s i, per tant, afegint Δ_x i calculant s' i Δ_f pot trencar la seguretat amb probabilitat 1.

Tanmateix, modificant aquesta idea obtindrem la seguretat forta que desitgem. Aquesta nova construcció va ser proposada a [3].

Teorema 5.11. *Considerem $\mathbb{F}$ un cos de mida q i característica p i d un enter tal que $p \nmid d+2$, definim*

$$\begin{aligned} \mathcal{E}: \mathbb{F}^d &\to \mathbb{F}^d \times \mathbb{F} \times \mathbb{F} \\ s &\mapsto (s, x, f(x,s)) \end{aligned}$$

on $x \in_R \mathbb{F}$ però ara

$$f(x,s) = x^{d+2} + \sum_{i=1}^{d} s_i x^i$$

*Aquesta construcció és un $(q^d, q^{d+2}, (d+1)/q)$ AMD-*code *sistemàtic amb seguretat forta i amb $\varpi = 2\log(q)$.*

Demostració. Tant les dimensions del codi com el valor d'ϖ es comproven fàcilment. Així doncs, només ens falta argumentar el valor $\delta = (d+1)/q$.

Volem veure que per a qualsevol $s \in \mathbb{F}$ i $\Delta \in \mathbb{F}^{d+2}$ tenim que $\Pr[\mathcal{D}(\mathcal{E}(s)+\Delta) \notin \{s, \perp\}] \leq \delta$. Ens serà suficient demostrar que per a qualsevol $s \neq s'$ i qualssevol $\Delta_x, \Delta_f \in \mathbb{F}$, $\Pr[f(x,s) + \Delta_f = f(x+\Delta_x, s')] \leq \delta$. Considerem doncs el següent succés

$$x^{d+2} + \sum_{i=1}^{d} s_i x^i + \Delta_f = (x+\Delta_x)^{d+2} + \sum_{i=1}^{d} s'_i (x+\Delta_x)^i$$

Reescrivim ara la part de la dreta de l'equació

$$(x+\Delta_x)^{d+2}+\sum_{i=1}^{d} s_i'(x+\Delta_x)^i = x^{d+2}+(d+2)\Delta_x x^{d+1}+\sum_{i=1}^{d} s_i' x^i + \Delta_x p(x)$$

on $p(x)$ és un polinomi de grau com a molt d. Usant aquesta nova expressió podem cancel·lar els termes x^{d+2} i ens queda

$$-(d+2)\Delta_x x^{d+1}+\sum_{i=1}^{d}(s_i'-s_i)x^i - \Delta_x p(x)+\Delta_f = 0$$

Ara veurem que aquesta expressió és un polinomi no nul de grau com a molt $d+1$. Ho farem considerant dues possibilitats. Si $\Delta_x \neq 0$, el coeficient de grau $d+1$ és $-(d+2)\Delta_x \neq 0$ ja que $d+2$ no és divisible per la característica del cos. Si $\Delta_x = 0$ el polinomi que ens queda és $\sum_{i=1}^{d}(s_i'-s_i)x^i+\Delta_f$ que no és nul perquè estem suposant $s \neq s'$.

Aquest argument posa de manifest que aquest polinomi té com a molt $d+1$ solucions per a x. Si anomenem B al conjunt de solucions llavors

$$\Pr[\mathcal{D}(\mathcal{E}(s)+\Delta) \notin \{s,\perp\}] = \Pr_{x\leftarrow\mathbb{F}}[x\in B] \leq \frac{d+1}{q}$$

□

Fixem-nos que aquest AMD-*code* és molt proper a l'òptim que havíem trobat anteriorment $\delta \geq (k-1)/n$. Si considerem el codi corrector d'errors que resulta d'aquesta construcció, veiem que és un codi molt semblant al de Reed Solomon però del qual hem n'eliminat les constants i el terme de grau $d+1$. A més a més hem considerat sempre el coeficient principal igual a 1. Amb aquestes accions aconseguim que el codi no sigui ni lineal ni cíclic (ja que prenem un representant de cada classe). Aquest codi té dimensió d, per tant l'òptim s'assoliria per a $\delta = (d-1)/n$ i no per a $\delta = (d+1)/n$ que és el que aconseguim amb aquesta construcció. De totes maneres aquesta construcció no dista molt d'aquesta fita i el valor de la *tag size* efectiva també és molt bo.

Corol·lari 5.12. *La* tag size *efectiva d'aquest AMD-*code *és* $\varpi^*(\kappa,u) \leq 2\kappa + 2\log(u/\kappa+3)+2$.

Amb aquesta construcció per a AMD-*codes*, si usem el Teorema 5.9 per obtenir KMS-MACs, aconseguirem una família de KMS-MACs molt propers a l'òptim perquè per a qualssevol $\kappa, u \in \mathbb{N}$, tindrem un (S,G,T,δ) KMS-MAC amb $\delta \leq 2^{-\kappa}$ i $S \geq 2^u$ tal que

$$\log(G) \leq 2\kappa + 2\log(u/\kappa+3)+2$$
$$\log(T) \leq \kappa + 2\log(u/\kappa+3)+1$$

Aquesta construcció per als AMD-*codes* és més flexible que les primeres en el sentit que la probabilitat d'error ja no és $1/|\mathcal{S}|$ (que en aquest cas seria $1/q^d$) sinó que és $\delta = (d+1)/q$ aconseguint així que la *tag size* i la *tag size* efectiva tinguin valors molt més desitjables. Això es tradueix en una major llibertat per poder incrementar la mida del missatge de sortida sense que haguem d'afegir més redundància de la necessària.

De totes maneres si volem augmentar la mida del missatge d mantenint la probabilitat d'error δ haurem d'augmentar q és a dir, haurem de canviar el cos base. És per aquest motiu que presentem una construcció més recent que no té aquest problema.

Hem vist una aplicació que permet obtenir AMD-*codes* amb seguretat forta a partir d'AMD-*codes* amb seguretat dèbil amb l'ajuda d'un codi d'autentificació (Teorema 5.6).

En aquest cas usarem un codi corrector d'errors per suplir el MAC. Usarem l'AMD-*code* amb seguretat dèbil escalable proposat per [9] i explicat anteriorment en la secció 5.6. Recordem que

$$\begin{aligned} \mathcal{E}: \mathbb{F}_{q^r} &\to \mathbb{F}_{q^r} \times \mathbb{F}_q \\ x &\mapsto (x, \phi(x^2)) \end{aligned}$$

és un $(q, q^{r+1}, 1/q)$ AMD-*code* amb seguretat dèbil on $\phi(s)$ és un epimorfisme $\phi : \mathbb{F}_{q^r} \to \mathbb{F}_q$.

Ara considerem $k = q^r$ i $n = q^{r+1}$, posem $\mathcal{S} = \mathbb{F}_q^k$ i definim un codi lineal $[n, k, d]$ amb $C \subset \mathbb{F}_q^n$. Si ara prenem $\mathcal{G}_1 = \mathbb{F}_q^{r+1}$ i $\mathcal{G}_2 = \mathbb{F}_q^2$ podem definir l'AMD-*code* amb seguretat forta següent:

$$\begin{aligned} \mathcal{E}: \mathcal{S} &\to \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \\ s &\mapsto (s, x, (\phi(x^2) \cdot (c(s))_x)) \end{aligned}$$

L'estratègia de l'adversari en aquest cas no es pot basar en canviar x, ja que el terme $\phi(x^2)$ (és a dir, la utilització de l'AMD-*code* amb seguretat dèbil) ens ho impedeix. Així doncs, ja no ens hem de preocupar de les rotacions del codi corrector. De totes maneres haurem de continuar procurant que el codi usat no tingui paraules amb molts símbols iguals, és a dir hauríem de treballar amb el codi a $\mathbb{F}_{q^n}/\langle 1, \ldots, 1\rangle$ i adaptar la distància en aquest nou espai per aconseguir-ne valors alts (en principi quan fem el pas al quocient la distància disminueix i ens serà més difícil aconseguir valors elevats d'aquesta).

Per altra banda si considerem el codi corrector adequat podem arribar a una probabilitat d'error de $n - d/n = \alpha k/n = \alpha/q$ amb α constant. Aquesta construcció és més flexible en el sentit que si volem augmentar la mida del missatge (és a dir incrementar la k) haurem d'augmentar la r (que també comporta augmentar la n). Però en cap cas cal canviar la q del cos base, que és el que passava en la construcció anterior. Només hauríem de redefinir una nova $\phi(x)$ i reajustar el codi a utilitzar (del qual, si en tenim una construcció genèrica, no ens hauria de comportar massa problemes).

Bibliografia

[1] S.Cabello, C.Padró, G.Sáez. Secret sharing schemes with detection of cheaters for a general acces structure. *Designs, Codes and Cryptography* **25** (2002), 175–188.

[2] A.Cavoukian, A.Stoianov. Biometric Encryption: A Positive-Sum Technology that Achieves Strong Authentication, Security and Privacity. Discusion Paper from Information & Privacy Comissioner of Ontario. Avaliable at www.ipc.on.ca.

[3] R.Cramer, Y.Dodis, S.Fehr, C.Padró, D.Wichs. Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors. *Advances in Cryptology - EUROCRYPT'08. Lecture Notes in Computer Science* **4965** (2008), 471–488.

[4] Y.Dodis, J.Katz, L.Reyzin, A.Smith. Robust Fuzzy Extractors and Authenticated Key Agrement from Close Secrets. *Advances in Cryptology - CRYPTO'06. Lecture Notes in Computer Science.* **4117** (2006), 232–250.

[5] Y.Dodis, R.Ostrovsky, L.Reyzin, A.Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.* **38** (2008), 97–139.

[6] Y.Dodis, D.Wichs. Robust Fuzzy Extractors Revisited. Preprint (2007).

[7] B.Kanukurthi, L.Reyzin. An Improved Robust Fuzzy Extractor. *Security and Cryptography for Networks. Lecture Notes in Computer Science* **5229** (2008), 156–171.

[8] S.Obana, T.Araki. Optimum secret sharing scheme secure against Cheating for Arbitrary Secret Distribution. *Advances in Cryptology - ASIACRYPT'06. Lecture Notes in Computer Science.* **4284** (2006), 364–379

[9] W.Ogata, K.Kurosawa. Optimum secret sharing schemes secure against cheating. *Advances in Criptology - EUROCRYPT'96 Lecture Notes in Computer Science* **1070** (1996), 200–211.

[10] S.Roman. *Coding and Information Theory.* Graduate Texts in Mathematics. Springer-Verlag, 1992.

[11] A.Sahai, B.Waters. Fuzzy Identity-Based Encryption. *Advances in Cryptology - EUROCRYPT'05. Lecture Notes in Computer Science* **3494** (2005), 457–473.

[12] W.J. Scheirer, T.E. Boult. Cracking Fuzzy Vaults and Biometric Encryption. In *Procedings Biometric Symposium, 2007*, pp 1–6.

[13] C.Soutar, D.Roberge, A.Stoianov, R.Gilroy, V.Kumar. Biometric Encryption^{TM}. Internal Technical Report, Bioscrypt Inc.

Adreces web d'interès:

[14] biolab.csr.unibo.it/Home.asp

[15] www.lgiris.com/ps/technology/index.htm

[16] www.innovatrics.com/technology/algorithm/

[17] www.biometric-fingerprint.com/fingerprint-live-finger-detection.html

[18] www.abomem.com.tw/fingerprint.htm

[19] www.kimaldi.com/area_de_conocimiento/biometria/encriptacion_biometrica

[20] www.tempresas.cl/revista/edicion_01/pdf/008-017%20Biometria.pdf

[21] www.turismo.uma.es/turitec/turitec2002/actas/Microsoft%20Word%20-%207.MARAVAL.pdf

[22] www.microsiervos.com/archivo/curiosidades/huellas-dactilares.html

[23] www.newscientist.com/article.ns?id=mg18725174.500

[24] www.cio.com/article/32296/Criminal_Justice_Fed_Fingerprint_Database_Spreads_Across_U.S.

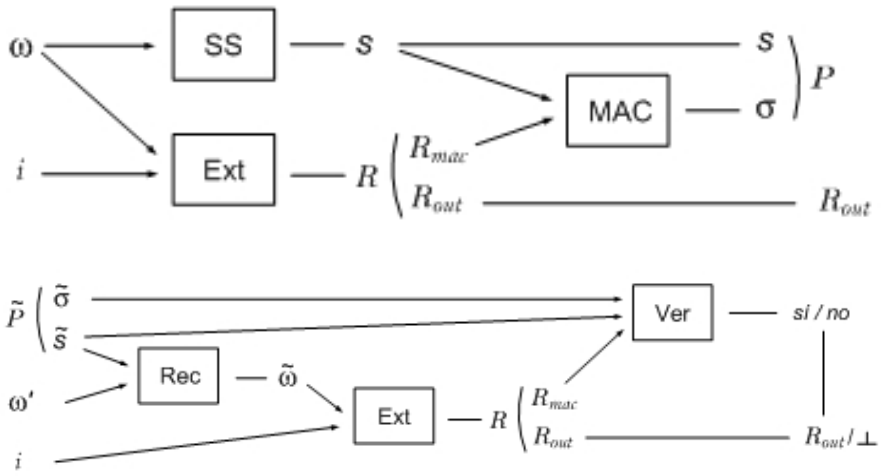
[25] www.nitgen.com/new_site/eng/sol/sol_finger.asp

[26] www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-62302007000100002&lng=es&nrm=iso

[27] www.dnielectronico.es

Fe d'errates

- pàg.19 línia 1 – Considerem X i Y variables aleatòries.
- pàg.22 línia 15 – $\mathbf{Rep}(\omega', \tilde{P}) \neq \perp$.
- pàg.24 – *Els esquemes 2.1 i 2.2 correctes són:*



- pàg.49 línia 10 – $\mathcal{G}_{23} = [23, 12, 7]$. *La taula 4.4 correcta és:*

codi	**blocs**	**bits totals**	**cap. correctora**	**% màxim d'errors**
$[23, 12, 7]$	11	253	3	13,04 %
$[24, 12, 8]$	11	264	3	12,5 %

- pàg.61 línia 27 – $\delta \geq \Pr[\mathcal{E}(s) + \Delta \in \cup_{s \neq s'} \mathcal{D}^{-1}(s')] = \frac{|\cup_{s \neq s'} \mathcal{D}^{-1}(s')|}{G-1} \geq \frac{(S-1)/\delta}{G-1}$.
- pàg.65 línia 12 – Per tant la probabilitat que $\mathcal{D}(e + \Delta) = s + \Delta_s$ és com a molt $\delta' + p_S$.