

Coding theory as a tool in crypto

Thomas Johansson

Dept of EIT,
Lund University,
Sweden

Indocrypt 2009

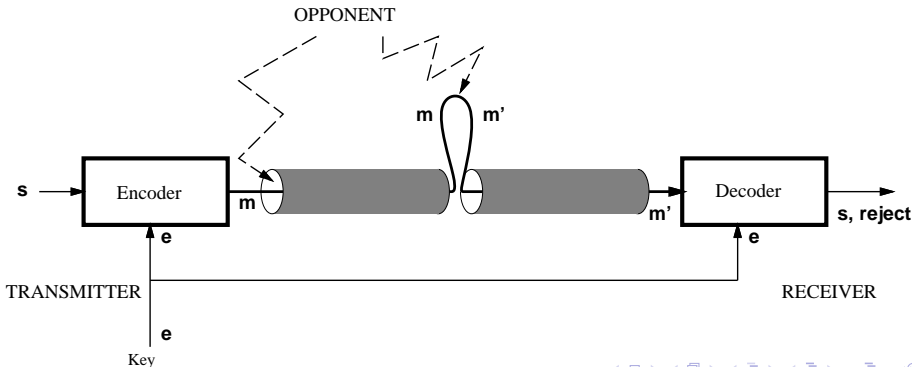
Coding theory has many applications in cryptography.

- PART I: Authentication codes
- PART II: Correlation attacks on stream ciphers

Unconditionally secure authentication

An unconditionally secure solution

- Simmons' model (Gilbert, MacWilliams, Sloane 1974)
- The transmitted information is a *source message*, s from \mathcal{S} .
- mapped into a (channel) *message*, $m \in \mathcal{M}$.
- the secret *key*, e and taken from the set \mathcal{E} .



Unconditionally secure authentication

- Encoding

$$f : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{M}, \quad (s, e) \mapsto m.$$

If $f(s, e) = m$ and $f(s', e) = m$, then $s = s'$ (injective for each $e \in \mathcal{E}$).

- The mapping f together with \mathcal{S} , \mathcal{M} and \mathcal{E} define an *authentication code* (A-code).
- The receiver must check whether a source message s exists, such that $f(s, e) = m$.
- If such an s exists, m is accepted (m is called valid).
- Otherwise, m is not authentic and thus rejected.

The opponent has two possible attacks at his disposal:

- The *impersonation attack*: Inserting a message m and hoping for it to be accepted as authentic.
- *substitution attack*: opponent observes the message m and replaces this with another message m' , $m \neq m'$, hoping for m' to be valid.

Definitions of attack success

The opponent chooses the message that maximizes his chances of success when performing an attack.

- Success in impersonation attack:

$$P_I = \max_m P(m \text{ is valid})$$

- Success in substitution attack:

$$P_S = \max_{\substack{m, m' \\ m \neq m'}} P(m' \text{ is valid} | m \text{ is valid}).$$

Probability of deception P_D as $P_D = \max(P_I, P_S)$.

Theorem

For any authentication code,

$$P_I \geq \frac{|\mathcal{S}|}{|\mathcal{M}|},$$
$$P_S \geq \frac{|\mathcal{S}| - 1}{|\mathcal{M}| - 1}.$$

$|\mathcal{M}|$ must be chosen much larger than $|\mathcal{S}|$.

Theorem (Simmons' bounds)

For any authentication code,

$$\begin{aligned} P_I &\geq 2^{-I(M;E)}, \\ P_S &\geq 2^{-H(E|M)}, \quad \text{if } |\mathcal{S}| \geq 2. \end{aligned}$$

For a good protection, i.e., P_I small, we must give away a lot of information about the key.

The square root bound

Multiply the two bounds together and get

$$P_I P_S \geq 2^{-I(M;E) - H(E|M)} = 2^{-H(E)}.$$

From $H(E) \leq \log |\mathcal{E}|$ we obtain the *square root bound*.

Theorem (Square root bound)

For any authentication code,

$$P_D \geq \frac{1}{\sqrt{|\mathcal{E}|}}.$$

On the square root bound

Theorem

The square root bound can be tight only if

$$|\mathcal{S}| \leq \sqrt{|\mathcal{E}|} + 1.$$

a large source size demands a twice as large key size. This is not very practical.

Systematic authentication codes

- An A-code for which the map $f : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{M}$ can be written in the form

$$f : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{S} \times \mathcal{Z}, \quad (s, e) \mapsto (s, z),$$

where $s \in \mathcal{S}, z \in \mathcal{Z}$, is called a *systematic* (or Cartesian) A-code.

- The second part z in the message is called the *tag* (or authenticator) and is taken from the tag alphabet \mathcal{Z} .

Systematic authentication codes

Theorem

For any systematic A-code

$$P_S \geq P_I.$$

Constructing authentication codes

Define $\mathcal{E}(m)$ as the set of keys for which a message m is valid,

$$\mathcal{E}(m) = \{e \in \mathcal{E}; \exists s \in \mathcal{S}, f(s, e) = m\}.$$

The probability of success in a substitution attack can be written as

$$P_S = \max_{\substack{m, m' \\ m \neq m'}} \frac{|\mathcal{E}(m) \cap \mathcal{E}(m')|}{|\mathcal{E}(m)|},$$

provided that the keys are uniformly distributed.

The vector space construction:

- Let $|\mathcal{S}| = q$, $|\mathcal{Z}| = q$, and $|\mathcal{E}| = q^2$, q prime power.
- Decompose the key as $e = (e_1, e_2)$, where $s, z, e_1, e_2 \in \mathbb{F}_q$.
- For transmission of source message s , generate a message $m = (s, z)$, where

$$z = e_1 + se_2.$$

Theorem

The above construction provides $P_I = P_S = 1/q$. Moreover, it has parameters $|\mathcal{S}| = q$, $|\mathcal{Z}| = q$, and $|\mathcal{E}| = q^2$.

$$\begin{aligned} P_S &= \max_{\substack{m, m' \\ m \neq m'}} \frac{|\mathcal{E}(m) \cap \mathcal{E}(m')|}{|\mathcal{E}(m)|} \\ &= \max_{\substack{m, m' \\ m \neq m'}} \frac{|\{e \in \mathcal{E}; m = (s, e_1 + se_2), m' = (s', e_1 + s'e_2)\}|}{|\{e \in \mathcal{E}; m = (s, e_1 + se_2)\}|} \\ &= \max_{\substack{m, m' \\ m \neq m'}} \frac{|\{e \in \mathcal{E}; m = (s, e_1 + se_2), m - m' = (s - s', (s - s')e_2)\}|}{|\{e \in \mathcal{E}; m = (s, e_1 + se_2)\}|} \\ &= \frac{1}{q}. \end{aligned}$$

Polynomial evaluation construction

- Let $\mathcal{S} = \{\mathbf{s} = (s_1, \dots, s_k); s_i \in \mathbb{F}_q\}$. Define the source message polynomial to be

$$s(x) = s_1x + s_2x^2 + \dots + s_kx^k.$$

- Let $\mathcal{E} = \{e = (e_1, e_2); e_1, e_2 \in \mathbb{F}_q\}$ and $\mathcal{Z} = \mathbb{F}_q$.
- For the transmission of source message \mathbf{s} , the transmitter sends \mathbf{s} together with the tag

$$z = e_1 + s(e_2).$$

Theorem

The construction gives systematic A-codes with parameters

$$|\mathcal{S}| = q^k, \quad |\mathcal{E}| = q^2, \quad |\mathcal{Z}| = q, \quad P_I = 1/q, \quad P_S = k/q.$$

$$\begin{aligned} P_S &= \max_{\substack{m, m' \\ m \neq m'}} \frac{|\mathcal{E}(m) \cap \mathcal{E}(m')|}{|\mathcal{E}(m)|} \\ &= \max_{\substack{s, s', z, z' \\ s \neq s'}} \frac{|\{e \in \mathcal{E}; z = e_1 + s(e_2), z' = e_1 + s'(e_2)\}|}{|\{e \in \mathcal{E}; z = e_1 + s(e_2)\}|} \\ &= \max_{\substack{s, s', z, z' \\ s \neq s'}} \frac{|\{e \in \mathcal{E}; z = e_1 + s(e_2), z - z' = s(e_2) - s'(e_2)\}|}{|\{e \in \mathcal{E}; z = e_1 + s(e_2)\}|} \\ &= \frac{k}{q}. \end{aligned}$$

Authentication codes using coding theory

- Let $m = (s, z)$ and write

$$z = e(s),$$

i.e., every key describes a map $\mathcal{S} \rightarrow \mathcal{Z}$.

- Let $n = |\mathcal{E}|$, $M = |\mathcal{S}|$ and $q = |\mathcal{Z}|$.
- Write $\mathcal{E}(s, z) = \{e \in \mathcal{E} : e(s) = z\}$.
- We restrict to A-codes for which

$$|\mathcal{E}(m)| = |\mathcal{E}| / |\mathcal{Z}| = n/q, \quad \forall m \in \mathcal{M}.$$

or $P_I = 1/q$.

Interpret the A-code as a code

- $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$.
- A q -ary code C of length $|\mathcal{E}|$ with $|\mathcal{S}|$ codewords by

$$\mathbf{c}^{(i)} = (c_1^{(i)}, c_2^{(i)}, \dots, c_n^{(i)})$$

where

$$c_j^{(i)} = e_j(s_i).$$

	e_1	e_2	e_3	\dots	e_n
s_1	$(c_1^{(1)})$	$c_2^{(1)}$	$c_3^{(1)}$	\dots	$c_n^{(1)}$
s_2	$(c_1^{(2)})$	$c_2^{(2)}$	$c_3^{(2)}$	\dots	$c_n^{(2)}$
\vdots	\vdots	\vdots		\ddots	\vdots
s_M	$(c_1^{(M)})$	$c_2^{(M)}$	$c_3^{(M)}$	\dots	$c_n^{(M)}$

Interpret the A-code as a code

-

$$C = \{\mathbf{c}^{(i)}; i = 1, 2, \dots, M\}.$$

- Define

$$\gamma(\mathbf{c}, \mathbf{c}') = \max_{\alpha, \beta \in \mathcal{Z}} |\{j; c_j = \alpha, c'_j = \beta\}|$$

- Define the *a-distance*

$$D(\mathbf{c}, \mathbf{c}') = n - q\gamma(\mathbf{c}, \mathbf{c}').$$

- Define the *minimum a-distance*

$$D(C) = \min_{\mathbf{c}, \mathbf{c}' \in C; \mathbf{c} \neq \mathbf{c}'} D(\mathbf{c}, \mathbf{c}').$$

- Triangle inequality does not hold!

Interpret the A-code as a code

$$P_S = \max_{\substack{s,z,s',z' \\ s \neq s'}} \frac{|\mathcal{E}(s,z) \cap \mathcal{E}(s',z')|}{|\mathcal{E}(s,z)|} = \max_{\substack{\mathbf{c}, \mathbf{c}' \\ \mathbf{c} \neq \mathbf{c}'}} \max_{\alpha, \beta \in \mathcal{Z}} \frac{|\{j; c_j = \alpha, c'_j = \beta\}|}{|\{j; c_j = \alpha\}|}$$

- Relation

$$D(C) = n(1 - P_S).$$

Trivially, we have $D(C) \leq d_H(C)$, where $d_H(C)$ is the minimum (Hamming) distance of C .

Bounds on authentication codes

- $m(n, \epsilon, q)$ maximal number of source messages in an A-code with n keys, $P_S \leq \epsilon$, tag size q .
- $A_q(n, d)$ maximal number of codewords in a q -ary code of length n and minimum Hamming distance d
- $A_q^*(n, d)$ the same but assuming equal symbol composition.

Theorem

We have

$$q(q-1)m(n, \epsilon, q) \leq A_q^*(n, (1-\epsilon)n),$$

and

$$q(q-1)m(n, \epsilon, q) + q \leq A_q(n, (1-\epsilon)n).$$

Bounds on authentication codes

We prove

$$q(q-1)m(n, \epsilon, q) = A_q^*(n, (1-\epsilon)n),$$

by constructing

$$C' = \{c'; c' = ac + b\mathbf{1}, c \in C, a \neq 0, b \in \mathbb{F}_q\}.$$

and then argue that $d_H(C') \geq D(C)$.

Bounds on authentication codes

Theorem

Any systematic A -code for which $P_I = P_S = 1/q$ satisfies

$$(q - 1) |\mathcal{S}| \leq n - 1.$$

- The code will have parameters

$$(n, M, d) = (n, q(q - 1) |\mathcal{S}| + q, \theta \cdot n),$$

where $\theta = 1 - P_S = (q - 1)/q$.

- The Plotkin bound gives

$$A_q(n, \theta n) \leq qA_q(n - 1, \theta n) \leq q \frac{\theta n}{\theta n - \theta(n - 1)} = qn.$$

Bounds on authentication codes

Theorem

Any systematic A-code satisfies

$$P_S \geq \frac{\log_q((q-1)|\mathcal{S}|+1)}{n}.$$

- Apply the Singleton bound $n \geq d + k - 1$, $k = \log_q M$.

Bounds on authentication codes

The previous result can be strengthened.

Theorem

Any systematic A-code satisfies

$$P_S \geq \frac{q \lfloor \log_q |\mathcal{S}| \rfloor}{n},$$

provided $\log_q |\mathcal{S}| < \sqrt{(2n(1 - 1/q)/q)} - 1/2$.

- The proof is more complicated and involves the Johnson bound on *binary constant weight codes*.

Going the other way: Constructing A-codes from codes

Theorem

Let a code C with parameters (n, M, d) be given, with the special property that if $\mathbf{c} \in C$ then $\mathbf{c} + \lambda \mathbf{1} \in C$, for all $\lambda \in \mathbb{F}_q$. Then there exists an A-code with parameters

$$|\mathcal{S}| = Mq^{-1}, |\mathcal{E}| = nq, P_I = 1/q, P_S = 1 - d/n.$$

- 1. Form a “quotient code” $C/\mathbf{1}$ with parameters $(n, M/q, d)$.
- 2. Expand \mathbf{c} to length nq by

$$\mathbf{c} \mapsto (\mathbf{c}, \mathbf{c} + \alpha_1 \mathbf{1}, \mathbf{c} + \alpha_2 \mathbf{1}, \dots, \mathbf{c} + \alpha_{q-1} \mathbf{1}).$$

Going the other way: Constructing A-codes from codes

- For a linear code the condition if $\mathbf{c} \in C$ then $\mathbf{c} + \lambda \mathbf{1} \in C$, for all $\lambda \in \mathbb{F}_q$ simply means

$$\mathbf{1} \in C$$

- Finding such codes we can now construct A-codes...

Existence bounds on A-codes

- $A_q^\bullet(n, d)$ maximal number of codewords in a q -ary code of length n and minimum Hamming distance d such that if $\mathbf{c} \in C$ then $\mathbf{c} + \lambda \mathbf{1} \in C$, for all $\lambda \in \mathbb{F}_q$.
- Modified Gilbert bound:

$$A_q^\bullet(n, d) \geq \frac{q^n}{V_q(n, d-1)},$$

where $V_q(n, d-1) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i$ is the size of a Hamming sphere.

Theorem

The maximal number of source messages satisfies

$$m(nq, \epsilon, q) \geq \frac{q^{n-1}}{V_q(n, (1-\epsilon)n-1)},$$

where $P_S \leq \epsilon$.

Constructions of A-codes

- An $[n = q, k + 1, d]$ Reed-Solomon code C over \mathbb{F}_q can be described as

$$C = \{(f(0), f(\alpha_1), f(\alpha_2), \dots, f(\alpha_{q-1})); f \in L\},$$

where L is the set of all polynomials of degree $< k + 1$ in $\mathbb{F}_q[x]$ and $\mathbb{F}_q = \{0, \alpha_1, \dots, \alpha_{q-1}\}$.

- $\mathbf{1} \in C$ so apply the construction. We get:

Polynomial evaluation construction

Let $\mathcal{S} = \{\mathbf{s} = (s_1, \dots, s_k); s_i \in \mathbb{F}_q\}$. Define the source message polynomial to be $s(x) = s_1x + s_2x^2 + \dots + s_kx^k$. Let $\mathcal{E} = \{e = (e_1, e_2); e_1, e_2 \in \mathbb{F}_q\}$ and $\mathcal{Z} = \mathbb{F}_q$. For the transmission of source message \mathbf{s} , the transmitter sends \mathbf{s} together with the tag

$$z = e_1 + s(e_2).$$

Theorem

The construction gives systematic A-codes with parameters

$$|\mathcal{S}| = q^k, \quad |\mathcal{E}| = q^2, \quad |\mathcal{Z}| = q, \quad P_I = 1/q, \quad P_S = k/q.$$

More on the polynomial evaluation construction

- An A-code is *weakly optimal* if for fixed $|\mathcal{S}|, |\mathcal{E}|, |\mathcal{Z}|, P_I$ we have P_S at its lowest value.

Theorem

The polynomial evaluation construction gives weakly optimal A-codes.

Recall

$$P_S \geq \frac{q \lfloor \log_q |\mathcal{S}| \rfloor}{n},$$

Constructions through concatenation of codes

- Very large source sizes ($\log |\mathcal{S}| = 2^{30}$) requires very large codes.
- We can get that by concatenating codes, for example RS codes.

Construction: Let $Q = 2^{r+s}$ and $q = 2^r$. The source message is a polynomial $s(x) \in \mathbb{F}_Q[x]$ of degree $\leq 2^s$. Let $e_1, e_2 \in \mathbb{F}_Q$ and $e_3 \in \mathbb{F}_q$. The tag is

$$z = e_3 + [s(e_1)e_2],$$

where $[x]$ returns the last r bits of x .

Parameters:

$$\log |\mathcal{S}| = (r + s)(1 + 2^s), \quad \log |\mathcal{E}| = 3r + 2s, \quad P_S < 2/2^r.$$

More recent developments

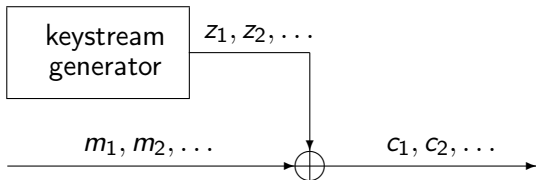
Focus on efficient implementation rather than minimum key size.

- LFSR-based Hashing and Authentication (Krawczyk)
- Bucket hashing (Rogaway)
- UMAC (Black, Halevi, Krawczyk, Krovetz, Rogaway)
- The Poly1305 MAC (Bernstein)
- and others...

In standards we have

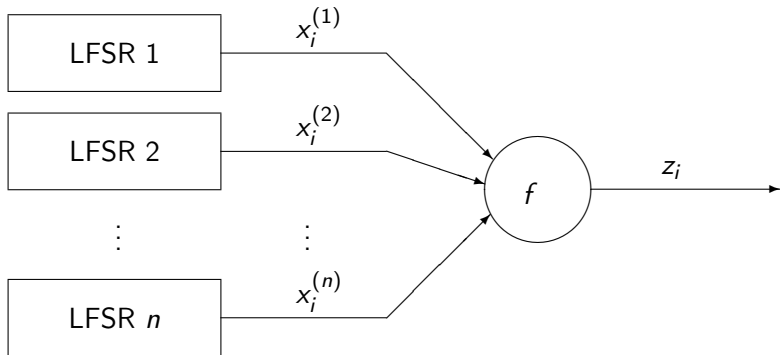
- NISTs GCM mode
- ETSI (3GPP) UIA2 mode.

PART II: Correlation attacks on stream ciphers



- The keystream generator contains one or several LFSRs.
- Observed keystream sequence z_1, z_2, \dots, z_N .

Correlation attacks



A nonlinear combining generator

Siegenthaler introduced *correlation attacks*.

A coding theory problem

- l is the length of the LFSR.
- The set of all 2^l possible LFSR sequences $\mathbf{u} = u_1, u_2, \dots, u_N$ form a linear $[N, l]$ code, call it C .
- Assume that we know N binary keystream symbols

$$\mathbf{z} = z_1, z_2, \dots, z_N.$$

- Then \mathbf{z} corresponds to a received word, obtained by sending an unknown codeword through the BSC.
- Our problem is to *decode* \mathbf{z} to the correct codeword.
- Typical characteristics:
 - Code length N is very large.
 - The noise is very strong (p close to $1/2$).

Meier-Staffelbach original approach

- Better than exhaustively searching LFSR – fast correlation attacks.
- Assume a *low weight* of $g(x)$.

Finding parity checks:

- The feedback polynomial

$$g(x) = 1 + g_1x^1 + g_2x^2 + \dots + g_lx^l.$$

- t = the number of taps, i.e., the weight of $g(x)$ is $t + 1$.
- Recurrence relation

$$u_n = g_1u_{n-1} + g_2u_{n-2} + \dots + g_lu_{n-l}.$$

- We get in this way $t + 1$ different parity check equations for u_n .

Fast correlation attacks

- $g(x)^j = g(x^j)$ for $j = 2^i$, low degree multiples of $g(x)$.
- We create new weight $t + 1$ parity checks by

$$g_{k+1}(x) = g_k(x)^2.$$

- This squaring is continued until the degree of $g_k(x)$ is greater than the length N of the observed keystream.
- Each $g_k(x)$ gives $t + 1$ new parity check equations for a fixed position u_n .

Fast correlation attacks

- Write m equations for position u_n as,

$$\begin{aligned}u_n + b_1 &= 0, \\u_n + b_2 &= 0, \\&\vdots \\u_n + b_m &= 0,\end{aligned}$$

where each b_i is the sum of t different positions of \mathbf{u} .

- Applying the same to the keystream

$$\begin{aligned}z_n + y_1 &= L_1 \\z_n + y_2 &= L_2 \\&\vdots \\z_n + y_m &= L_m.\end{aligned}$$

where y_i is the sum of the positions in the keystream corresponding to the positions in b_i .

Fast correlation attacks

- Assume that h out of the m equations hold, i.e.,
 $h = |\{i : L_i = 0, 1 \leq i \leq m\}|$.
- Then it is possible to calculate the probability

$$p^* = P(u_n = z_n | h \text{ equations hold})$$

as

$$p^* = \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}},$$

where $p = P(u_n = z_n)$, and $s = P(b_i = y_i)$.

Fast correlation attacks

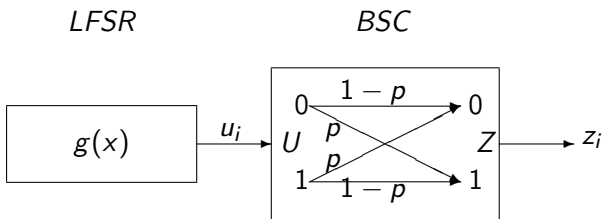
Algorithm B: the probabilities are calculated iteratively.

Two parameters p_{thr} and N_{thr} .

1. For all symbols in the keystream, calculate p^* and determine the number of positions N_w with $p^* < p_{thr}$.
2. If $N_w < N_{thr}$ repeat step 1 with $P(u_i = z_i) = p$ replaced by $P(u_i = z_i) = p^*$.
3. Complement the bits with $p^* < p_{thr}$ and reset the probabilities to p .
4. If not all equations are satisfied go to step 1.

Resembles iterative decoding a lot.

Correlation attacks



- General case: $g(x)$ is not of low weight.
- How to efficiently decode the “LFSR code” when transmitted over a very noisy BSC?

Correlation attacks using convolutional codes

- Idea: Find parity checks that include u_n , an arbitrary linear combination of u_{n-1}, \dots, u_{n-B} , together with at most t other symbols. say $t = 2$.

-

$$\begin{aligned}u_n + \sum_{i=1}^B c_{i1} u_{n-i} + b_1 &= 0, \\u_n + \sum_{i=1}^B c_{i2} u_{n-i} + b_2 &= 0, \\&\vdots \\u_n + \sum_{i=1}^B c_{im} u_{n-i} + b_m &= 0,\end{aligned}$$

where $b_k = \sum_{i=1}^{\leq t} u_{j_{ik}}$, $1 \leq k \leq m$ is the sum of (at most) t positions in \mathbf{u} .

Correlation attacks using convolutional codes

- We get a rate $R = \frac{1}{m+1}$ bi-infinite convolutional code V .
- For $v_n^{(i)}$ we create an estimate from \mathbf{z} .
- $v_n^{(0)} = u_n$ and $P(v_n^{(0)} = z_n) = 1 - p$. Otherwise, if $v_n^{(i)} = u_{j_1} + u_{j_2}$ then

$$P(v_n^{(i)} = z_{j_1} + z_{j_2}) = (1 - p)^2 + p^2.$$

- Using these estimates we can construct a sequence

$$\mathbf{r} = \dots r_n^{(0)} r_n^{(1)} \dots r_n^{(m)} r_{n+1}^{(0)} r_{n+1}^{(1)} \dots r_{n+1}^{(m)} \dots,$$

where $r_n^{(0)} = z_n$ and $r_n^{(i)} = z_{j_{1i}} + z_{j_{2i}}$, $1 \leq i \leq m$, that plays the role of a received sequence for the convolutional code.

Correlation attacks using convolutional codes

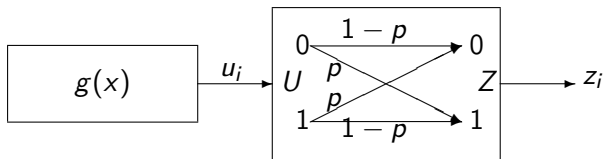
- To recover the initial state of the LFSR we need to decode l consecutive information bits correctly.
- Optimal decoding (ML decoding) of convolutional codes uses the Viterbi algorithm.
- There is neither a starting state, nor an ending state. Start by assigning the metrics $\log P(\mathbf{s} = z_1, z_2, \dots, z_B)$ to each state \mathbf{s} in the trellis. We then proceed to decode from $n = B$ as usual.

Another coding based approach

(Chepychov, Johansson, Smeets, FSE 2000)

LFSR

BSC



- The LFSR code \mathcal{C} has the $(N \times l)$ -generator matrix

$$G = \begin{pmatrix} h_1^1 & h_2^1 & \cdots & h_N^1 \\ h_1^2 & h_2^2 & \cdots & h_N^2 \\ \vdots & & \cdots & \\ h_1^l & h_2^l & \cdots & h_N^l \end{pmatrix}.$$

Another coding based approach

- Fix a $k < l$. Look for pairs of columns of G such that,

$$h_i^{k+1} = h_j^{k+1}, \dots, h_i^l = h_j^l, \quad 1 \leq i \neq j \leq N.$$

- The indices of all such pairs: $\{i_1, j_1\}, \dots, \{i_{n_2}, j_{n_2}\}$.
- The sum $c_i + c_j$ is independent of $u_{k+1}, u_{k+2}, \dots, u_l$,

$$c_i + c_j = (h_i^1 + h_j^1) u_1 + \dots + (h_i^k + h_j^k) u_k.$$

- This means that the words

$$(C_1, C_2, \dots, C_{n_2}) = (c_{i_1} + c_{j_1}, c_{i_2} + c_{j_2}, \dots, c_{i_{n_2}} + c_{j_{n_2}})$$

form an $[n_2, k]$ -code, referred to as \mathcal{C}_2 .

Another coding based approach

- Calculate

$$Z_1 = z_{i_1} + z_{j_1}, \dots, Z_{n_2} = z_{i_{n_2}} + z_{j_{n_2}},$$

a word acting as a received word for \mathcal{C}_2 .

- Decode the code \mathcal{C}_2 using exhaustive search through all the 2^k codewords of \mathcal{C}_2 .
- New much worse error probability
 $p_2 = P(C_i \neq Z_i) = 2p(1 - p)$, but dimension is smaller.

Precomputation.

- Choose a $k < l$ and a $t \geq 2$. Construct generator matrix G .
- Find all sets of t indices $\{i(1), i(2), \dots, i(t)\}$ that satisfy

$$\sum_{j=1}^t h_{i(j)}^m = 0, \text{ for } m = k + 1, k + 2, \dots, l.$$

Store the indices $i(1), i(2), \dots, i(t)$ together with the value of

$$\left(\sum_{j=1}^t h_{i(j)}^1, \sum_{j=1}^t h_{i(j)}^2, \dots, \sum_{j=1}^t h_{i(j)}^k \right).$$

(Parity checks for the code \mathcal{C}_t)

The general attack

Decoding.

Input: The received vector (z_1, z_2, \dots, z_N) .

Step 1. Compute

$$(Z_1 = \sum_{j=1}^t z_{i_1(j)}, \dots, Z_n = \sum_{j=1}^t z_{i_n(j)}).$$

Step 2. Decode the code \mathcal{C}_t using exhaustive search through the all 2^k codewords of \mathcal{C}_t .

Theorem

With given $k, l, t, p = 1/2 - \varepsilon$, the required length N of the observed sequence \mathbf{z} for the attack to succeed is

$$N \approx 1/4 \cdot (2kt! \ln 2)^{1/t} \cdot \varepsilon^{-2} \cdot 2^{\frac{l-k}{t}},$$

assuming $N \gg n_t$.

Correlation attacks: more recent attacks

- Canteaut and Trabbia: Use parity check equations of weight 4 and 5, decode with Gallager iterative decoding algorithm.
- Johansson, Jönsson: reconstruction of linear polynomials
- Mihaljevic, Fossorier and Imai: combine exhaustive search over the first B bits with list decoding algorithm.
- Chose, Joux and Mitton: new methods for efficient implementations, better decoding algorithm.
- Golic: vectorial approach to fast correlation attacks
- and many more

- We saw two examples of coding in crypto:
 - authentication codes
 - correlation attacks
- Many other: Boolean functions/S-boxes; McEliece PKC etc.;
- New application areas may come...