Cryptography Via Linear Codes

Nate Black

Clemson University

Master's Project Presentation

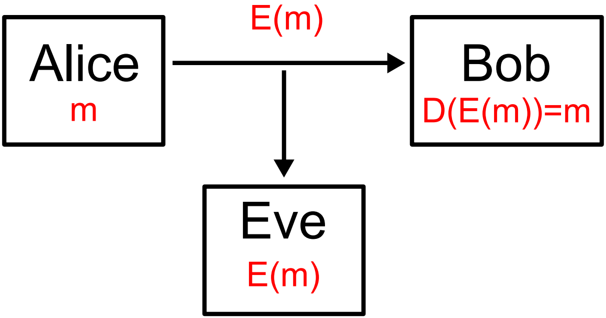
April 16, 2010

Outline

- 1. Background for Cryptography
- 2. Linear Codes
- 3. Decoding Linear Codes
- 4. Applications

Background

Cryptography Model



Background

Secret Key Cryptography

- Alice and Bob share the same key.
- **Advantage:** These methods are very secure.
- **Disadvantage:** Alice and Bob must have agreed on the key ahead of time.

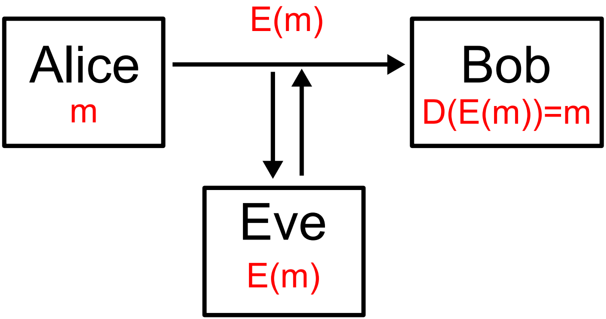
Background

Public Key Cryptography

- Alice and Bob have different keys.
- Bob publishes a Public Key which Alice uses to send Bob messages.
- Bob uses a Private Key to decode messages sent to him.
- **Advantage:** These methods provide secure communication without shared knowledge prior to communication.
- **Disadvantage:** Slight increase in overhead and computational complexity over secret key methods.

Background

Cryptography Model Revisited



Background

Active Attacks

- Chosen plaintext:
  Eve chooses several plaintexts and receives the corresponding ciphertexts.
- Chosen ciphertext:
  Eve chooses several ciphertexts and receives the corresponding plaintexts.

Background

Active Attacks

- Chosen plaintext:
  Eve chooses several plaintexts and receives the corresponding ciphertexts.
- Chosen ciphertext:
  Eve chooses several ciphertexts and receives the corresponding plaintexts.

Background

Cryptographic Primitives

- Integer Factorization (RSA)
  Factor a number into a product of primes: $n = p_1 p_2 \dots p_k$.
- Discrete Log Problem (ElGamal)
  Let a and b be elements of a finite field $\mathbb{F}$, then find $x \in \mathbb{F}$ such that $a^x = b$.
- Solving systems of polynomial equations
- The lattice problem
- The decoding problem from Coding Theory

Background

Cryptographic Primitives

- Integer Factorization (RSA)
  Factor a number into a product of primes: $n = p_1 p_2 \dots p_k$.
- Discrete Log Problem (ElGamal)
  Let a and b be elements of a finite field $\mathbb{F}$, then find $x \in \mathbb{F}$ such that $a^x = b$.
- Solving systems of polynomial equations
- The lattice problem
- The decoding problem from Coding Theory

Background

Cryptographic Primitives

- Integer Factorization (RSA)
  Factor a number into a product of primes: $n = p_1 p_2 \dots p_k$.
- Discrete Log Problem (ElGamal)
  Let a and b be elements of a finite field $\mathbb{F}$, then find $x \in \mathbb{F}$ such that $a^x = b$.
- Solving systems of polynomial equations
- The lattice problem
- The decoding problem from Coding Theory

Background

Cryptographic Primitives

- Integer Factorization (RSA)
  Factor a number into a product of primes: $n = p_1 p_2 \ldots p_k$.
- Discrete Log Problem (ElGamal)
  Let a and b be elements of a finite field $\mathbb{F}$, then find $x \in \mathbb{F}$ such that $a^x = b$.
- Solving systems of polynomial equations
- The lattice problem
- The decoding problem from Coding Theory

Background

Cryptographic Primitives

- Integer Factorization (RSA)
  Factor a number into a product of primes: $n = p_1 p_2 \dots p_k$.
- Discrete Log Problem (ElGamal)
  Let a and b be elements of a finite field $\mathbb{F}$, then find $x \in \mathbb{F}$ such that $a^x = b$.
- Solving systems of polynomial equations
- The lattice problem
- The decoding problem from Coding Theory

Definition

Linear Codes

Definition

- It is an error correcting code.
- The vectors in the code are called codewords.
- The codewords form a linear subspace.

Definition

- It is an error correcting code.
- The vectors in the code are called codewords.
- The codewords form a linear subspace.

Definition

- It is an error correcting code.
- The vectors in the code are called codewords.
- The codewords form a linear subspace.

Definition

Definition (Linear Code)

An $[n, k, d]$ linear code, C, is a k dimensional subspace of $\mathbb{F}^n$ where $\mathbb{F}$ is a field, and d is the minimum distance of the code. The elements $\mathbf{u} \in C$ are called codewords.

Definition (Minimum Distance)

Let $H(\mathbf{u}, \mathbf{v})$ be the Hamming distance between two codewords, where the Hamming distance between $\mathbf{u}$ and $\mathbf{v}$ is the number of positions in which $\mathbf{u}$ and $\mathbf{v}$ differ. Then the minimum distance, d, of a code, C, is given by $d = min(\{H(\mathbf{u}, \mathbf{v}) \mid \mathbf{u} \neq \mathbf{v} \text{ and } \mathbf{u}, \mathbf{v} \in C\})$.

Definition

- **Encoding**:
  Given $\mathbf{u} \in \mathbb{F}^k$ produce the corresponding codeword, $\mathbf{v} = \mathbf{u}G$.
- **Decoding**:
  Given $\mathbf{w} \in \mathbb{F}^n$ find the closest codeword, $\mathbf{c} \in C$.

The Generator Matrix

Let $\{\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_k\}$ be a basis for C, the k dimensional subspace of $\mathbb{F}^n$, where

$$\mathbf{c}_i = (c_{i,1},\ c_{i,2},\ \ldots,\ c_{i,n})$$

is an n-vector. Then define the $k \times n$ matrix G as follows:

$$G = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_k \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k,1} & c_{k,2} & \cdots & c_{k,n} \end{bmatrix}.$$

This matrix is called the generating matrix for C since $C = \left\{vG \mid v \in \mathbb{F}^k\right\}$ (i.e. all $\mathbb{F}$-linear combinations of the rows of G).

The Parity Check Matrix

Another related matrix which can be used to define the code C is the $(n-k) \times n$ matrix H of rank $n-k$ called the parity check matrix. This matrix is the solution to the following matrix equation:

$$GH^{\mathsf{T}} = 0_{k \times (n-k)}.$$

The Parity Check Matrix

Note that since every codeword, $\mathbf{v}$, can be written as $\mathbf{u}G = \mathbf{v}$ this implies that

$$\mathbf{v}H^{\mathsf{T}} = \mathbf{u}GH^{\mathsf{T}} = \mathbf{u}0_{k\times(n-k)} = 0_{1\times(n-k)}.$$

Also, since the rank of H is $n-k$ and $\{\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_k\} \subseteq C$ is a linearly independent set of size k with $\mathbf{c}_i H^{\mathsf{T}} = 0_{1\times(n-k)}$ we conclude that C is precisely the left null space of H^{T} and thus we have the following useful property:

$$\mathbf{v}H^{\mathsf{T}} = 0_{1\times(n-k)} \text{ iff } \mathbf{v} \in C.$$

Practical Applications

- ISBN codes
- CDs
- Space probe photographs
- RAID arrays

Reed-Solomon Codes

- Let $\mathbb{F}$ be a field of size q, and $1 \leq k \leq n \leq q$
- $A = \{\alpha_1, \alpha_2, \ldots \alpha_n\} \subseteq \mathbb{F}$ with $\alpha_i \neq \alpha_j$ is called the evaluation set
- $Z = (z_1,\ z_2,\ \ldots,\ z_n)$ with $z_i \neq 0 \in \mathbb{F}$ are called the scaling coefficients
- Codewords: $\mathbf{c}_i = \left(z_1\alpha_1^{i-1},\ z_2\alpha_2^{i-1},\ \ldots,\ z_n\alpha_n^{i-1}\right)$
- Minimum distance: $n - k + 1$

Reed-Solomon Codes

The Generator Matrix:

$$G = G_1 Z$$

$$G = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{bmatrix} \begin{bmatrix} z_1 & 0 & \dots & 0 \\ 0 & z_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z_n \end{bmatrix}$$

- $det(Z) \neq 0$ since $z_i \neq 0$.
- G_1 is a Vandermonde matrix.
- If $k = n$, then $\det(G_1) = \prod_{1 \leq i < j \leq n} (\alpha_j - \alpha_i) \neq 0$ since $\alpha_i \neq \alpha_j$.

Reed-Solomon Codes

$$
(u_0,\ u_1,\ \ldots,\ u_{k-1}) \quad \cdot \quad \begin{bmatrix} z_1 & z_2 & \ldots & z_n \\ z_1\alpha_1 & z_2\alpha_2 & \ldots & z_n\alpha_n \\ z_1\alpha_1^2 & z_2\alpha_2^2 & \ldots & z_n\alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1\alpha_1^{k-1} & z_2\alpha_2^{k-1} & \ldots & z_n\alpha_n^{k-1} \end{bmatrix}
$$

$$
= \left(z_1 \sum_{i=0}^{k-1} u_i\alpha_1^i,\ z_2 \sum_{i=0}^{k-1} u_i\alpha_2^i,\ \ldots,\ z_n \sum_{i=0}^{k-1} u_i\alpha_n^i \right)
$$

$$
= (z_1 u(\alpha_1),\ z_2 u(\alpha_2),\ \ldots,\ z_n u(\alpha_n))
$$

Reed-Solomon Codes

Thus all the codewords in a Reed-Solomon code are simply the n-tuples of the form

$$(z_1 f(\alpha_1),\ z_2 f(\alpha_2),\ \ldots,\ z_n f(\alpha_n))$$

obtained by evaluating over all

$$f \in \mathbb{F}[x] \text{ with } deg(f) < k.$$

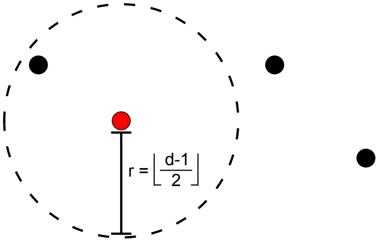
Decoding

Decoding Linear Codes

Unambiguous Decoding

Definition (Unambiguous Decoding)

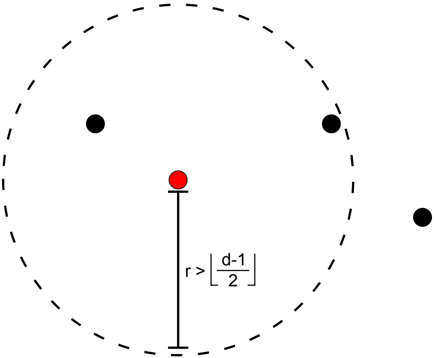
For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find the codeword, if it exists, within the ball of radius $r = \left\lfloor \frac{d-1}{2} \right\rfloor$ centered around $\mathbf{w}$.



List Decoding

Definition (List Decoding)

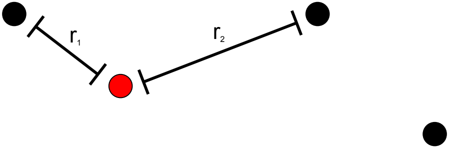
For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find all codewords, if any exist, within the ball of radius $r > \left\lfloor \frac{d-1}{2} \right\rfloor$ centered around $\mathbf{w}$.



Maximum Likelihood Decoding

Definition (Maximum Likelihood Decoding)

For an $[n, k, d]$ code and input $\mathbf{w} \in \mathbb{F}^n$, find the closest codeword to $\mathbf{w}$ with respect to the Hamming distance.



Decoding

Why Maximum Likelihood Decoding?

	vector components	distance
Received vector:	$[1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1]$	
Codeword 1:	$[1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]$	3
Codeword 2:	$[1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0]$	4

Reed-Solomon Decoding Problem

Definition (Reed-Solomon Decoding Problem)

Given n points: $\alpha_1, \alpha_2, \ldots \alpha_n$ in a finite field, $\mathbb{F}$, and a vector $\mathbf{u} = (u_1,\ u_2,\ \ldots,\ u_n) \in \mathbb{F}^n$ find $g \in \mathbb{F}[x]$ with $deg(g) < k$ such that for $\mathbf{v} = (f(\alpha_1),\ f(\alpha_2),\ \ldots,\ f(\alpha_n))$, $H(\mathbf{u}, \mathbf{v}) \leq H(\mathbf{u}, \mathbf{w})\ \ \forall\ \mathbf{w} \in C$ with $\mathbf{w} \neq \mathbf{u}, \mathbf{w} \neq \mathbf{v}$.

Maximum Likelihood Decoding of Reed-Solomon codes has been shown to be NP-hard, which means that these problems are excellent candidates for use in constructing cryptosystems.

Polynomial Reconstruction Problem

Definition (Polynomial Reconstruction Problem)

Let $\mathbb{F}$ be a finite field, and let n, k, and t be given design parameters. For a given set of n ordered pairs, $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\} \subseteq \mathbb{F}^2$ find all $f \in \mathbb{F}[x]$ with $deg(f) < k$ such that $f(x_i) = y_i$ for at least t indices, where $1 \leq t \leq n$. Oftentimes, a PR problem is represented as a 6-tuple: $(n, k, t, \mathbf{x}, \mathbf{y}, \mathbb{F})$, where $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_n)$ and $\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_n)$.

This problem is exactly the same as list decoding for a Reed-Solomon code.

Applications

Cryptosystems and an Application

McEliece Cryptosystem

A Binary Goppa code is used for the underlying security of this cryptosystem.

- **Private Key**:
  1. G, a $k \times n$ generator matrix for an $[n, k, d]$ Goppa code, C
  2. S, a nonsingular random $k \times k$ matrix sometimes known as a scrambler
  3. P, an $n \times n$ permutation matrix
- **Public Key**:
  1. The product of the private key matrices, $K = SGP$
  2. The number of errors, t, that C can correct

McEliece Cryptosystem

- **Encryption:**
  1. **Input:** a message $\mathbf{m} = (m_1,\ m_2,\ \ldots,\ m_k)$
  2. Compute
     $$\mathbf{c} = \mathbf{m}K + \mathbf{e} = \mathbf{m}SGP + \mathbf{e},$$
     where $\mathbf{e}$ is a random n-vector with $H(\mathbf{e}, \mathbf{0}) \leq t$.
- **Decryption:**
  1. **Input:** a received vector, $\mathbf{c} = (c_1,\ c_2,\ \ldots,\ c_n)$
  2. Compute
     $$\mathbf{c}P^{-1} = \mathbf{m}SG + \mathbf{e}P^{-1}.$$
  3. Now $\mathbf{m}SG$ is a codeword in C, and $\mathbf{e}P^{-1}$ has $H(\mathbf{e}P^{-1}, \mathbf{0}) \leq t$ so that by applying the decoding algorithm for C to $\mathbf{c}P^{-1}$ we obtain
     $$\mathbf{m}S.$$
  4. This allows us to multiply by S^{-1} on the right and obtain $\mathbf{m}$.

McEliece Cryptosystem

- Secure against known attacks.
- Dan Bernstein and some others recently broke the code with the original parameters using a cluster of 200 computers for a couple of weeks. However, their attack fails when the parameters are increased.
- The best known general attack uses a technique called information set decoding.

Information Set Decoding

- An information set, $I = \{i_1, i_2, \dots i_k\}$, is a size k subset of the indices of the columns from the $k \times n$ public key matrix, K, such that the reduced $k \times k$ matrix K_I formed from the columns specified in the information set is invertible.
- Similarly, for a vector $\mathbf{v} = (v_1,\ v_2,\ \dots,\ v_n)$, let $\mathbf{v}_I = (v_{i_1},\ v_{i_2},\ \dots,\ v_{i_k})$ be the reduced version of $\mathbf{v}$ formed by the entries in $\mathbf{v}$ having indices in I.

Information Set Decoding Algorithm

Given a received vector $\mathbf{c} = \mathbf{m}K + \mathbf{e}$ perform the following steps:

1. Approximate $\mathbf{m}$ by $\mathbf{u} = \mathbf{c}_I K_I^{-1}$
2. Calculate the codeword $\mathbf{v} = \mathbf{u}K$
3. If $H(\mathbf{v} - \mathbf{c}) \leq t$ then $\mathbf{m} = \mathbf{u}$ otherwise choose another information set and run the algorithm again

Note that $\mathbf{v}_I = (\mathbf{mK})_I$ if and only if the indices in I were not corrupted by errors in the encryption process. To break the McEliece system the attacker runs this algorithm on all information subsets until $\mathbf{m}$ is found. In practice the attacker will not know which information sets do not contain errors so he will try all possible information sets.

Niederreiter Cryptosystem

This cryptosystem is a modification of the McEliece system and as originally proposed, used a GRS code for security.

- **Private Key:**
  1. H, an $r \times n$ parity check matrix for a GRS code, C
  2. S, a nonsingular random $r \times r$ matrix sometimes known as a scrambler
- **Public Key:**
  1. The product of the private key matrices, $K = SH$
  2. r

Niederreiter Cryptosystem

- **Encryption:**
  1. **Input:** a message $\mathbf{m} = (m_1, m_2, \ldots, m_n)$
  2. Compute
  $$\mathbf{c} = \mathbf{m}K^\mathsf{T} = \mathbf{m}H^\mathsf{T}S^\mathsf{T},$$
  Note that $\mathbf{m}$ should have $H(\mathbf{m}, \mathbf{0}) < \lfloor \frac{n-((n-r)-1)}{2} \rfloor = \lfloor \frac{r+1}{2} \rfloor$, otherwise there will be too many errors to uniquely recover the plaintext, $\mathbf{m}$, from the ciphertext $\mathbf{c}$.
- **Decryption:**
  1. **Input:** a received vector, $\mathbf{c} = (c_1, c_2, \ldots, c_r)$
  2. Compute
  $$\mathbf{w} = \mathbf{c}(S^\mathsf{T})^{-1} = \mathbf{m}H^\mathsf{T}.$$
  3. Then, using the efficient decoding algorithm for C, recover $\mathbf{m}$.

Sidelnikov-Shestakov attack

- The goal of the attack is to factor K into the product of two trapdoor matrices H_{tr} and S_{tr} such that $K = SH = H_{tr}S_{tr}$.
- H_{tr} and H should both be parity check matrices for the same GRS code, and S_{tr} should be an invertible $n \times n$ matrix.
- If K can be decomposed into such a product then the cryptosystem is broken since we can perform the following steps to recover m from a recieved vector c:

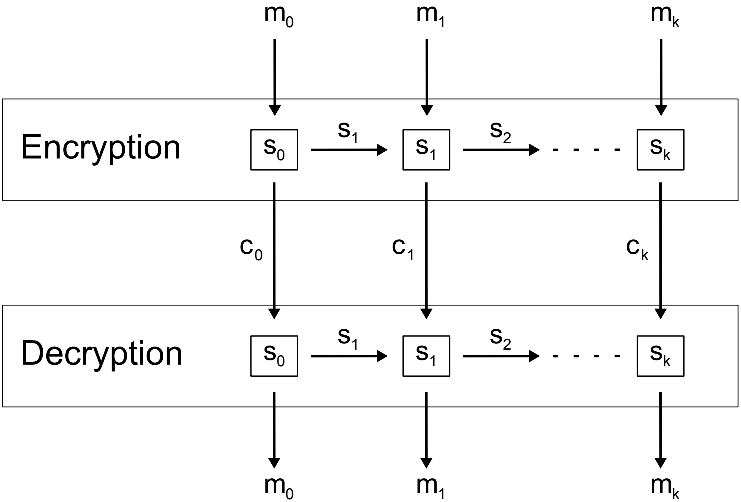
$$\begin{aligned} c &= mK^{\mathsf{T}} \\ c &= mS_{tr}^{\mathsf{T}}H_{tr}^{\mathsf{T}} \end{aligned}$$

- Use the decoding algorithm to recover mS_{tr}^{T}.
- Then multiply by $(S_{tr}^{\mathsf{T}})^{-1}$ on the right to obtain m.

PR Based Stateful Cipher

- The cipher is said to be a block cipher since the plaintext message is broken up into blocks to be encrypted.
- It is a stateful cipher since the encryption of each block depends on the current state of the encryption algorithm.
- It exhibits forward security, since the blocks are encrypted one after another in a chain, so that if one of the blocks in the chain is decrypted then the security fails for all remaining blocks, but the previous blocks remain secure.
- It can be implemented as a secret key cryptosystem.

PR Based Stateful Cipher



Setup

- An instance of the PR problem, $(n, k, t, \mathbf{z}, \mathbf{y}, \mathbb{F})$, having $z_i \neq 0 \;\; \forall\, i$ and $z_i \neq z_j \;\; \forall\, i \neq j$
- $K = \{s \in \mathbb{F}_2^n \mid H(s, 0) = t\}$, that is the set of all n-bit strings having exactly t ones in their representation
- I_s denotes the size t subset of $\{1, 2, \ldots n\}$ corresponding to the indices of $s \in K$ that are ones
- b_s is the integer with binary representation s
- Pick a random $s_0 \in K$ as the initial state (i.e. the secret key) which is known by both the sender and the receiver

Encryption

Input: A state, $s \in K$, and a message block, $\mathbf{m} \in \mathbb{F}^{\frac{k-1}{2}}$

1. Generate the next state by picking a random $s' \in K$.
2. Define a polynomial $p(x) \in \mathbb{F}[x]$ with $deg(p) < k$ by interpolating the following k points where r_i are random elements of $\mathbb{F}$.

$$\left\{ \begin{array}{ll} (0, b_{s'}) & \\ (z_i, m_i) & i = 1, 2, \ldots \dfrac{k-1}{2} \\ (z_i, r_i) & i = \dfrac{k-1}{2} + 1, \dfrac{k-1}{2} + 2, \ldots k-1 \end{array} \right\}$$

3. Generate an error vector, $\mathbf{e}$ as follows:

$$e_j = 0 \quad \forall\, j \in I_s \qquad e_j = r_j \quad \forall\, j \notin I_s$$

where r_j are random elements of $\mathbb{F}$.

4. Return the encrypted vector, $\mathbf{c} \in \mathbb{F}^n$, given by

$$\mathbf{c} = (p(z_1),\ p(z_2),\ \ldots,\ p(z_n)) + \mathbf{e}.$$

Decryption

Input: A state, $s \in K$, and a received encrypted block, $\mathbf{c} \in \mathbb{F}^n$

1. Interpolate the set of t points

$$\{(z_i, c_i) \mid i \in I_s\}$$

to obtain $f(x)$ with $deg(f) < k$. Note that none of these points were corrupted by the error vector since $e_i = 0 \ \forall\, i \in I_s$.

2. Update the state of the algorithm to $f(0)$.
3. Return the recovered message, $\mathbf{m} \in \mathbb{F}^{\frac{k-1}{2}}$, given by

$$\mathbf{m} = \left(f(z_1),\ f(z_2),\ \ldots,\ f\left(z_{\frac{k-1}{2}}\right)\right)$$

PR Based Biometric Authentication

Definition (PR Problem Variation)

Let $\mathbb{F}$ be a finite field and m be a design parameter. For a given set of m 4-tuples, $\{(x_1, y_1, \bar{x}_1, \bar{y}_1), (x_2, y_2, \bar{x}_2, \bar{y}_2), \ldots (x_m, y_m, \bar{x}_m, \bar{y}_m)\} \subseteq \mathbb{F}^4$ with $x_1, x_2, \ldots x_m, \bar{x}_1, \bar{x}_2, \ldots \bar{x}_m$ nonzero and distinct find a polynomial $f \in \mathbb{F}[x]$ with $deg(f(x)) < m$ such that for each $1 \leq i \leq m$, $f(x_i) = y_i$ or $f(\bar{x}_i) = \bar{y}_i$.

Setup

- Let $v \in \mathbb{F}$ be the value that is to be secured.
- Select m of the users features to measure when authenticating, and let t_i be the average value among the user population for the ith chosen feature.
- Define $g_i : \mathbb{N} \to \mathbb{R}$, where $g_i(n) = r_n$ represents the measured value, r_n, for the feature, i, on the nth successful authentication. For example, if the measurement of the fourth feature on the eighth successful authentication attempt was 0.2, then $g_4(8) = 0.2$.
- A feature of a user is said to be distinguishing to the left (right) if the average value of the last h authentication attempts is statistically significant to the left (right) of the population average, t_i.

Calibration

1. Take h measurements of each of the users m features.
2. Choose a random polynomial $p \in \mathbb{F}[x]$ with $deg(p) < m$ such that $p(0) = v$.
3. Create a PR instance with the following set of tuples:

$$\begin{cases} (x_i, p(x_i), \bar{x}_i, r_i) & \text{if the } i\text{th feature is distinguishing to the left} \\ (x_i, r_i, \bar{x}_i, p(\bar{x}_i)) & \text{if the } i\text{th feature is distinguishing to the right} \\ (x_i, p(x_i), \bar{x}_i, p(\bar{x}_i)) & \text{otherwise} \end{cases}$$

where r_i is a random element of $\mathbb{F}$. This PR instance is then stored for use in the next authentication attempt.

Authentication

1. Let $\{(x_1, y_1, \bar{x}_1, \bar{y}_1), (x_2, y_2, \bar{x}_2, \bar{y}_2), \ldots (x_m, y_m, \bar{x}_m, \bar{y}_m)\}$ be the tuples in the PR instance.
2. Measure the users m features to generate the following set of ordered pairs:

$$\left\{ \begin{array}{ll} (x_i, y_i) & \text{if the } i\text{th feature is distinguishing to the left} \\ (\bar{x}_i, \bar{y}_i) & \text{if the } i\text{th feature is distinguishing to the right} \\ (x_i, y_i), (\bar{x}_i, \bar{y}_i) & \text{otherwise} \end{array} \right\}$$

3. Interpolating these ordered pairs should produce the original function $p(x)$, even if some of the features were measured with slight deviations (i.e. some of the wrong ordered pairs were included from the tuples in the PR instance).
4. If the user is successful in authenticating, then perform the calibration step again, using the average value of the last h successful authentication attempts for each feature.

Conclusion

Thank you for attending.