Error-Correcting Codes

Berlekamp-Welsh

**Source alphabet:**

Message: $S^n$

**Code:** encode $E: S^n \rightarrow \Sigma^k$,

$k = \text{length of codeword}$

$\Sigma = \text{alphabet}$

Decode: $D: \Sigma^k \rightarrow S^n$

$d = 1$,

properly: $E(f_{in}) = f_{out}$

1) erase & replace $S^n$ with $D(E(f_{in}))$

2) error, $e$ chosen

3) delete $f_{in}$, $f_{out}$ chosen

Message: $\Sigma^k \rightarrow \Sigma^n$

**Decoding:**

- Communication
- Storage: full-blank
- noisy code: bit flips
- bit flips on $e$ hard drive

Can we do better?

Yes: would $n \leq k$,

So sound to be decoded & map

**Code:**

$E: \Sigma^k \rightarrow \Sigma^n$

- correct errors in $f_{out}$

**Received:** $E(f_{in})$

Correct: $d$ errors in $E(f_{in})$

$\text{Min. dist.} = d$

$1)$ error (substitution)

Decide: map $E(f_{in})$ to $E(f_{out})$

$E(f_{in})$ contains $e$ errors.

Received: $E(f_{out})$

Correct: $d - 1$ errors in $E(f_{in})$

$2)$ error (substitution)

Decide: map $E(f_{in})$ to $E(f_{out})$

$E(f_{in})$ contains $e$ errors.

Received: $E(f_{out})$

Correct: $d - 1$ errors in $E(f_{in})$
\( N = \alpha \cdot k \) for \( \alpha = 0(1) \)
\( d = \beta \cdot N \) \( \beta = \Omega(1) \).

\( \Rightarrow \) can tolerate a constant fraction of failures.

E.g., take \( C = \) set of \( 2^k \) random vectors in \( \mathbb{F}_q^k \) (for \( q = 2, 9, 13 \)).

\[ E \] any map \( \begin{array}{c} \mathbb{F}_q^k \rightarrow C \\ 0 \end{array} \)

\( D \): decode to closest codeword.

Issue: how to compute \( E(D) \) efficiently, time \( \sim 2^k \), want's time \( \text{poly}(k) \).

Reed–Solomon codes

for \( \Sigma \) “large”.

\( \Sigma = \mathbb{F}_p \sim \) integer arithmetic modulo \( p \) prime.

Def: of the RS code:

\[ C = \{ f(x_1), f(x_2), \ldots, f(x_n) \} ; \]

\( f(x) = \sum_{i=0}^{k-1} x^i f_i \) \( \mathbb{F}_p \)

where \( \alpha_1, \ldots, \alpha_n \) are distinct in \( \mathbb{F}_p \).

an input message is encoded into a polynomial with coeff. \( = \) message.

\( (f(x), \ldots, f(x_n)) \rightarrow (f(x_1), f(x_2), \ldots, f(x_n)) \).

Parameters of the RS code?

think of \( n = \Theta(k) \).

Lem: min dist in RS code is \( d = n-k+1 \).

E.g., \( n = 2k \Rightarrow d = k+1 \)

\( \Rightarrow \) can tolerate \( k \) errors.

(2/2 sends).