Byzantine attackers- coding

- Use MDS-style codes
- Generalization due to recent work by Koetter and Kschischang
- Slides due to Sid Jaggi
Multicasting

Simplifying assumptions (for this talk)

Single source
Directed, acyclic graph.
Each link has unit capacity.
Links have zero delay.
Kinds of linearity

Algebraic codes

Block codes

Convolutional codes

\( b_1 b_2 \ldots b_m \rightarrow \alpha \)

\((b_1 b_2 \ldots b_m) \in \{0,1\}^m \rightarrow \alpha \in F(2^m)\)

\( \vec{\alpha} \in \{0,1\}^m \)

\( b_0 b_1 \ldots b_{m-1} \rightarrow \vec{\alpha} \)

\((b_0 b_1 \ldots b_{m-1}) \in \{0,1\}^m \rightarrow \vec{\alpha} \)

\( b_0 b_1 \ldots \rightarrow \alpha(z) \)

\((b_0 b_1 \ldots b_{m-1}) \in \{0,1\}^\infty \rightarrow \alpha(z) \in F_2(z)\)

\( \beta_1 \alpha_1 + \beta_2 \alpha_2 + \ldots + \beta_k \alpha_k \)

\([\beta_1] \alpha_1 + [\beta_2] \alpha_2 + \ldots + [\beta_k] \alpha_k \)

\( \beta_1(z) \alpha_1(z) + \ldots + \beta_k(z) \alpha_k(z) \)
Model 1 - Results

Model 1 - Encoding

C (Capacity)

p ("Noise parameter")
Model 1 - Encoding

\[ D_{ij} = T_j(1) \cdot 1 + T_j(2) \cdot r_i + \ldots + T_j(n(1-\varepsilon)) \cdot r_i^{n(1-\varepsilon)} \]
Model 1 - Encoding

\[ D_{ij} = T_j(1) \cdot 1 + T_j(2) \cdot r_i + \ldots + T_j(n(1-\varepsilon)) \cdot r_i^{n(1-\varepsilon)} \]
Model 1 - Transmission
Model 1 - Decoding

\[ D_{ij}' = T_j(1)' \cdot 1 + T_j(2)' \cdot r_i' + \cdots + T_j(n(1 - \varepsilon))' \cdot r_i'^{(n(1 - \varepsilon))} \]

"Quick consistency check"
Model 1 - Decoding

\[ D_{ij}' = T_j(1)' \cdot 1 + T_j(2)' \cdot r_i' + \ldots + T_j(n(1-\varepsilon))' \cdot r_i'^n(1-\varepsilon) ? \]

\[ D_{ji}' = T_i(1)' \cdot 1 + T_i(2)' \cdot r_j' + \ldots + T_i(n(1-\varepsilon))' \cdot r_j'^n(1-\varepsilon) ? \]

“Quick consistency check”
Model 1 - Decoding

Consistency graph

\[ D_{ij}' = T_j(1)' \cdot 1 + T_j(2)' \cdot r_i' + \ldots + T_j(n(1-\varepsilon))' \cdot r_i'^{n(1-\varepsilon)} \checkmark \]

\[ D_{ji}' = T_i(1)' \cdot 1 + T_i(2)' \cdot r_j' + \ldots + T_i(n(1-\varepsilon))' \cdot r_j'^{n(1-\varepsilon)} \checkmark \]

Edge i consistent with edge j
Model 1 - Decoding

Consistency graph

(Self-loops… not important)

\[
D_{ij} = T_j(1)' \cdot 1 + T_j(2)' \cdot r_i' + \ldots + T_j(n(1-\epsilon))' \cdot r_i'^{n(1-\epsilon)}
\]

\[
D_{ji} = T_i(1)' \cdot 1 + T_i(2)' \cdot r_j' + \ldots + T_i(n(1-\epsilon))' \cdot r_j'^{n(1-\epsilon)}
\]

Edge i consistent with edge j
Model 1 - Decoding

Consistency graph

Detection – select vertices connected to at least $|E|/2$ other vertices in the consistency graph.
Decode using $T_i$'s on corresponding edges.
Model 1 - Proof

Consistency graph

\[ \sum_k (T_j(k) - T_j(k')) \cdot r_i^k = 0 \]

Polynomial in \( r_i \) of degree \( n \) over \( F_q \),
value of \( r_i \) unknown to Zorba

Probability of error < \( n/q << 1 \)
Unicast

1. Code (X,Y,Z)
2. Message (X,Z)
3. Bad links (Z)
4. Coin (X)
5. Transmission (Y,Z)
6. Decode correctly (Y)

Eureka
Unicast

|E| directed unit-capacity links

Zorba (hidden to Xavier/Yvonne) controls |Z| links Z. p = |Z|/|E|

Xavier and Yvonne share no resources (private key, randomness)
Zorba computationally unbounded; Xavier and Yvonne can only perform “simple” computations.
Zorba knows protocols and already knows almost all of Xavier’s message (except Xavier’s private coin tosses)

Goal: Transmit at “high” rate and w.h.p. decode correctly
Background

- Noisy channel models (Shannon, ...)
  - Binary Symmetric Channel
Background

- Noisy channel models (Shannon, …)
  - Binary Symmetric Channel
  - Binary Erasure Channel

\[
C = \frac{1}{1-p}
\]
Background

- Adversarial channel models
  - "Limited-flip" adversary (Hamming, Gilbert-Varshamov, McEliece et al.)
  - Shared randomness, private key, computationally bounded adversary...

\[
p \quad (\text{"Noise parameter"})
\]

\[
C \quad (\text{Capacity})
\]
Unicast - Results

$p$ ("Noise parameter")

$C$ (Capacity)

$1-p$
Unicast - Results

p ("Noise parameter")

C (Capacity)

0

1

0.5

? ? ?
Unicast - Results

(Just for this talk, Zorba is causal)
Ignorant Zorba

1. Code (X,Y,Z)
2. Message $X_p,X_s (X)$
3. Bad links (Z)
4. Coin (X)
5. Transmission (Y,Z)
6. Decode correctly (Y,$\not\chi$)

$I(Z;X_s)=0$
\[ p = \frac{|Z|}{h} \]

General Multicast Networks

Slightly more intricate proof
Unicast - Encoding

Block-length $n$ over finite field $\mathbb{F}_q$

Vandermonde matrix

"Easy to use consistency information"
Unicast - Encoding

\[ D_i = T_i(1) + r + T_i(2) + r + \ldots + T_i(n(1 - \varepsilon)) + r^{n(1 - \varepsilon)} \]
Unicast - Transmission
Unicast - Quick Decoding

Choose majority \((r, D_1, \ldots, D_{|E|})\)

\[D_i = T_i(1).1 + T_i(2).r + \ldots + T_i(n(1-\epsilon)).r^{n(1-\epsilon)}\]  
If so, accept \(T_i\), else reject \(T_i\)

Use accepted \(T_i\)s to decode

\[\sum_k (T_i(k)-T_i(k')).r^k=0\]

Polynomial in \(r\) of degree \(n\) over \(F_q\),

value of \(r\) unknown to Zorba

Probability of error \(< n/q<<1\)
Observation: Can treat adversaries as new sources
General Multicast Networks

$y_i = T_i x$
General Multicast Networks

\[ y_i = T_i x \]
\[ y'_i = T_i x + T'_i a_i \]

\((x(1), x(2), \ldots, x(n))\) form a \(R\)-dimensional subspace \(X\)

\((a_i(1), a_i(2), \ldots, a_i(n))\) form a \(|Z|\)-dimensional subspace \(A_i\)

w.h.p. over network code design, \(TX\) and \(TA_i\) do not intersect (robust codes...).

w.h.p. over \(x(i), (y(1), y(2), \ldots y(R+|Z|))\) forms a basis for \(TX \oplus TA_i\)

But already know basis for \(TX\), therefore can obtain basis for \(TA_i\)
Variations - Feedback
Variations – Know thy enemy
Variations – Omniscient but not Omnipresent

Achievability: Gilbert-Varshamov, Algebraic Geometry Codes
Converse: Generalized MRRW bound
Variations – Random Noise
Ignorant Zorba - Results

\[ C (\text{Capacity}) = X_p + X_s \]

where:
- \( X_p \) is the noise parameter
- \( X_s \) is some other parameter

The graph shows a relationship between \( p \) ("Noise parameter") and \( C \) (Capacity). The equation \( 1 - 2p \) is also indicated on the graph.
Ignorant Zorba - Results

$X_p + X_s$

$X_s$

$C$ (Capacity)

$p$ (“Noise parameter”)

$P = 1 - 2p$

MDS code

$a + b + c$

$a + 2b + 4c$

$a + 3b + 9c$