Secure Network Coding via Filtered Secret Sharing

Jon Feldman, Tal Malkin, Rocco Servedio, Cliff Stein

(Columbia University)

{jonfeld@ieor, tal@cs, rocco@cs, cliff@ieor}.columbia.edu

Network Coding and Security

- Network coding: new model of transmission...
  - ...how do we make it secure?

1. Cai and Yeung[02]
   - wire-tap adversary: can look at any k edges.
   - Suff. conditions for $\exists$ secure multicast code.
2. Jain[04]: More precise cond. (one terminal).
3. Ho, Leong, Koetter, Médard, Effros, Karger [04]: Byzantine modification detection.

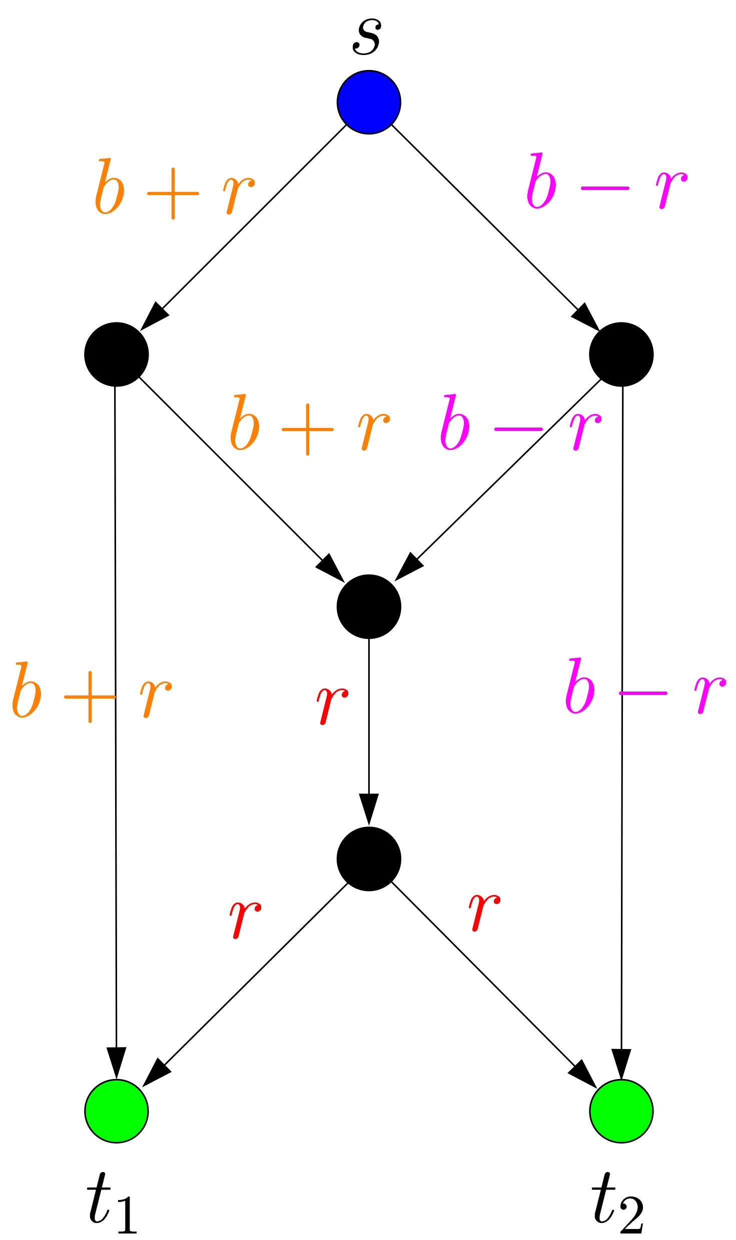
Network Coding and Security

- Network coding: new model of transmission...
  - ...how do we make it secure?

1. Cai and Yeung[02]
   - wire-tap adversary: can look at any k edges.
   - Suff. conditions for $\exists$ secure multicast code.
2. Jain[04]: More precise cond. (one terminal).
3. Ho, Leong, Koetter, Médard, Effros, Karger [04]: Byzantine modification detection.

- This talk: precise analysis of wire-tap adversary, balance between security, rate, edge bandwidth.
- Related: robustness [Koetter Médard 02].

Making our Example Secure



- Use $\mathbb{F}_3 = \{0, 1, 2\}$.
- Less ambitious goal: Send *one* symbol $b \in \mathbb{F}_3$ to both sinks.
- Choose $r \in \mathbb{F}_3$ randomly.
- Can define symbols s.t. any single wire-tapper learns nothing about b, both sinks can compute b.

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$. min-cut value = $n = \min_i \kappa_i$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$. min-cut value = $n = \min_i \kappa_i$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$. min-cut value = $n = \min_i \kappa_i$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.
- Feasibility of transmission:

  (i) Every $v[b, c]$ spanned by $\{v[a, b]\}_a$ (or $a = s$).

Linear Multicast Network Coding (No Security)

Given: Network $G = (V, E)$, source $s \in V$, sinks $T \subseteq V$. min-cut value = $n = \min_i \kappa_i$.

- Goal: get message $m \in \mathbb{F}_q^n$ to every sink.
- Network code:
  - Define coding vectors $v[e] \in \mathbb{F}_q^n$ for each edge.
  - Edge carries symbol $m \cdot v[e]$.
- Feasibility of transmission:

  (i) Every $v[b, c]$ spanned by $\{v[a, b]\}_a$ (or $a = s$).
- Recoverability at sinks:

  (ii) For all $t \in T$, the vectors $\{v[a, t]\}_a$ span $\mathbb{F}_q^n$.

Wire-Tap Model, Randomness at the Source

- Adversary has access to *any* set of k edges,
  - knows symbol transmitted along edge,
  - knows network code, topology,
  - has unlimited computational power.

Wire-Tap Model, Randomness at the Source

- Adversary has access to *any* set of k edges,
  - knows symbol transmitted along edge,
  - knows network code, topology,
  - has unlimited computational power.
- Source allowed to generate random symbols (r).
  - (Jain [04]: random bits at intermediate nodes)

Wire-Tap Model, Randomness at the Source

- Adversary has access to *any* set of k edges,
  - knows symbol transmitted along edge,
  - knows network code, topology,
  - has unlimited computational power.
- Source allowed to generate random symbols (r).
  - (Jain [04]: random bits at intermediate nodes)
- Task: design function $m = f(x, r)$ at source, coding vectors on edges s.t.:
  - Coding vectors satisfy feasibility,
  - Information x recoverable at each sink,
  - Information x secure against adversary.

Wire-Tap Model, Randomness at the Source

- Adversary has access to *any* set of k edges,
  - knows symbol transmitted along edge,
  - knows network code, topology,
  - has unlimited computational power.
- Source allowed to generate random symbols (r).
  - (Jain [04]: random bits at intermediate nodes)
- Task: design function $m = f(x, r)$ at source, coding vectors on edges s.t.:
  - Coding vectors satisfy feasibility,
  - Information x recoverable at each sink,
  - Information x secure against adversary.
- Goal: information-theoretic security.

Security, Rate and Bandwidth

- We study possible trade-offs between security, rate and bandwidth:

Security = k = # edges tapped $< n = \min_i \kappa_i$.

Rate = t = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

Security, Rate and Bandwidth

- We study possible trade-offs between security, rate and bandwidth:

Security = k = # edges tapped $< n = \min_i \kappa_i$.

Rate = t = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

- Easy to show: $t \leq n - k$.

Security, Rate and Bandwidth

- We study possible trade-offs between security, rate and bandwidth:

Security = k = # edges tapped $< n = \min_i \kappa_i$.

Rate = t = # information symbols multicast.

Edge Bandwidth = $\log q$, where symbols in $\mathbb{F}_q$.

- Easy to show: $t \leq n - k$.
- Cai and Yeung [02]: If $q > \binom{|E|}{k}$, can send $t = n - k$ symbols securely.
  - Construction time $\approx \binom{|E|}{k}$.

Our Results

- If you give up a little capacity, bandwidth requirement reduced significantly:

Thm: For any $c > 1$, if $q \geq |E|^{\Omega(\frac{1}{c-1})}$, can send $t = n - ck$ symbols securely.

- Algorithm: poly-time, secure w.h.p.
- If $k = \Theta(|E|)$, only need $q \geq 2^{\Omega(\frac{1}{c-1})}$.

Our Results

- If you give up a little capacity, bandwidth requirement reduced significantly:

  Thm: For any $c > 1$, if $q \geq |E|^{\Omega(\frac{1}{c-1})}$, can send $t = n - ck$ symbols securely.

  - Algorithm: poly-time, secure w.h.p.
  - If $k = \Theta(|E|)$, only need $q \geq 2^{\Omega(\frac{1}{c-1})}$.

- If you do not give up capacity, then bandwidth might have to be large:

  Thm: If $t = n - k$, then there are examples where all solutions (using this method) must have $q \gtrsim |E|^{\sqrt{k}}$.

Relation w/ Cai & Yeung

- Core Lemma of Cai and Yeung: If one can construct a matrix with certain independence properties relative to the coding vectors, then the network code can be altered to achieve security.

Relation w/ Cai & Yeung

- Core Lemma of Cai and Yeung: If one can construct a matrix with certain independence properties relative to the coding vectors, then the network code can be altered to achieve security.
- Our extensions:
  - Independence properties are also necessary.
  - Using orthogonal space, re-cast as coding theory problem.
  - Give up some capacity to make coding problem solvable.
  - Use necessary direction, covering radius, to prove negative result.
  - Observation: don't alter code, just input.

Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.

Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.

Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges, can learn nothing about x.

  (More formally, for all sets $E' \subset E$ with $|E'| \leq k$, If r is a random vector in $\mathbb{F}_q^\ell$, the random variable $(f(x, r) \cdot v[e])_{e \in E'}$ is independent of x.)

Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges, can learn nothing about x.

  (More formally, for all sets $E' \subset E$ with $|E'| \leq k$, If r is a random vector in $\mathbb{F}_q^\ell$, the random variable $(f(x, r) \cdot v[e])_{e \in E'}$ is independent of x.)
- Recoverability, feasibility follow immediately (as long as x can be determined from $f(x, r)$).

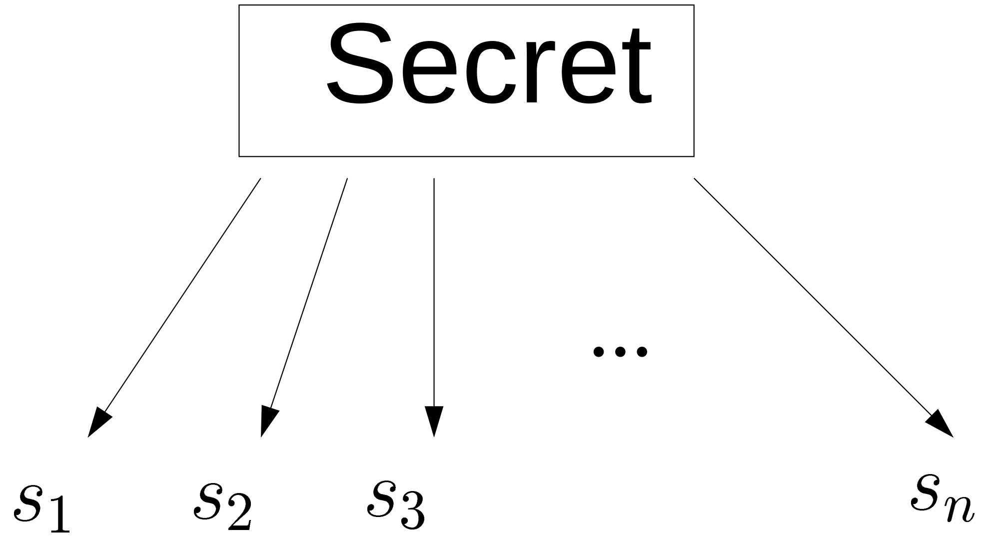
Making a Given Network Code Secure [CY02]

Given a linear solution $(v[e])_{e \in E}$ to a network coding problem, can we use it securely?

- Set $m = f(x, r)$, with info $x \in \mathbb{F}_q^t$, random $r \in \mathbb{F}_q^\ell$.
- Send message m normally using network code.
- Security condition: If adversary looks at any k edges, can learn nothing about x.

  (More formally, for all sets $E' \subset E$ with $|E'| \leq k$, If r is a random vector in $\mathbb{F}_q^\ell$, the random variable $(f(x, r) \cdot v[e])_{e \in E'}$ is independent of x.)
- Recoverability, feasibility follow immediately (as long as x can be determined from $f(x, r)$).
- Advantage: don't need to alter network code.

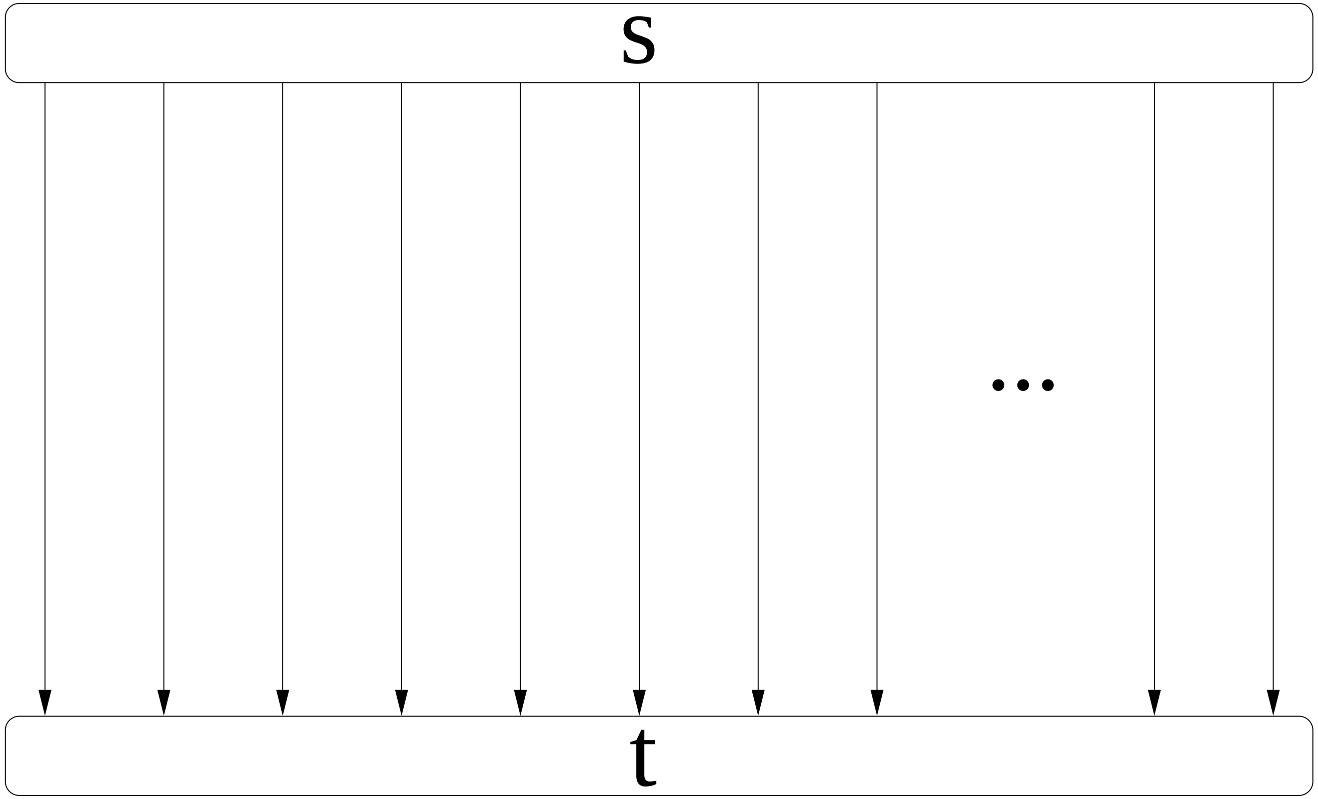
Secret Sharing (Shamir)



- Dealer has “secret”.
- Distribute shares $s_1, \ldots, s_n$ s.t.
  - Given any k shares, can recover secret.
  - Given any $k-1$ shares, learn nothing.
- Computational/info-theoretic security
- “Access pattern” for recoverability/security.

Connection to Secret Sharing

- Simple case: single source/sink, n parallel edges, adversary has any set of k edges.



- Modest goal $t = 1$: send one symbol $x \in \mathbb{F}_q$.

Connection to Secret Sharing

- Suppose $v[e_i]$ = $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$.

  $\iff f(x, r)$ is the “dealer” in secret sharing.

- Encoding:
  - Choose $(r_1, \ldots r_k)$ at random.
  - Let $p(z) = x + r_1 z + r_2 z^2 + \cdots + r_k z^k$
  - Set message $m = (p(\alpha_1), \ldots, p(\alpha_n))$.
  - (Encode (x,r) using a Reed-Solomon code.)

Connection to Secret Sharing

- Suppose $v[e_i]$ = $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$.

  $\iff$ $f(x, r)$ is the "dealer" in secret sharing.

- Encoding:
  - Choose $(r_1, \ldots r_k)$ at random.
  - Let $p(z) = x + r_1 z + r_2 z^2 + \cdots + r_k z^k$
  - Set message $m = (p(\alpha_1), \ldots, p(\alpha_n))$.
  - (Encode (x,r) using a Reed-Solomon code.)

- Decoding:
  - Knowing all $p(\alpha_i)$ reveals p (interpolation).
  - Knowing k or fewer m_i's tells you nothing.
  - Works for any $k \leq n - 1$.

Filtered Secret Sharing

- “Classical” secret sharing: adversary has k shares.

Filtered Secret Sharing

- “Classical” secret sharing: adversary has k shares.
- “Filtered” secret sharing:
  - “Menu” of N lin. combos of all n shares
  - Adversary chooses k items from the menu

Filtered Secret Sharing

- "Classical" secret sharing: adversary has k shares.
- "Filtered" secret sharing:
  - "Menu" of N lin. combos of all n shares
  - Adversary chooses k items from the menu
- Secret is $x \in \mathbb{F}_q^t$

Filtered Secret Sharing

- "Classical" secret sharing: adversary has k shares.
- "Filtered" secret sharing:
  - "Menu" of N lin. combos of all n shares
  - Adversary chooses k items from the menu
- Secret is $x \in \mathbb{F}_q^t$

- Given: n-by-N full-rank "filter" matrix V over $\mathbb{F}_q$.
- Find: Function $f : \mathbb{F}_q^t \times \mathbb{F}_q^\ell \to \mathbb{F}_q^n$ such that:

  For all n-by-k submatrices V' of V,

  over random $r \leftarrow \mathbb{F}_q^\ell$,

  we have $f(x, r) \cdot V'$ indep. of x $(\forall x)$.

Filtered Secret Sharing

- Given: n-by-N full-rank “filter” matrix V over $\mathbb{F}_q$.
- Find: Function $f : \mathbb{F}_q^t \times \mathbb{F}_q^\ell \to \mathbb{F}_q^n$ such that:

  For all n-by-k submatrices V' of V,

  over random $r \leftarrow \mathbb{F}_q^\ell$,

  we have $f(x, r) \cdot V'$ indep. of x.

- Classical: special case $t = 1$, $N = n$, $V = I$.
- For network coding:
  - $N = |E|$, V is n-by-$|E|$ matrix of coding vectors.
  - Ignores network topology.

Design of Linear Error-Correcting Codes

- Code C_G = linear subspace generated by the rows of β-by-N matrix G.
- Distance(C_G) = $\min_{y \in C_G} \Delta(y, 0^n)$.
- Goal in coding theory:
  - For given rate β/N, find code with large distance.

Design of Linear Error-Correcting Codes

- Code C_G = linear subspace generated by the rows of β-by-N matrix G.
- Distance(C_G) = $\min_{y \in C_G} \Delta(y, 0^n)$.
- Goal in coding theory:
  - For given rate β/N, find code with large distance.

- Generalization:
  - Designed code must be far from 0^n, and some other given points.

Generalized Coding Problem

- For generators A, B, define:

$$\Delta_s(A,B) \equiv \min_{y\in C_A,\ y'\in C_B,\ y'\neq \mathbf{0}} \Delta(y,y')$$

  - *Note: Asymmetric*
  - *Note:* $\Delta_s(A,B) \leq$ *min-dist(*C_B*)*

Generalized Coding Problem

- For generators A, B, define:

$$\Delta_s(A,B) \equiv \min_{y \in C_A,\ y' \in C_B,\ y' \neq \mathbf{0}} \Delta(y, y')$$

  - *Note: Asymmetric*
  - *Note:* $\Delta_s(A,B) \leq$ *min-dist(*C_B*)*

- **"Span Distance Problem"** (over field $\mathbb{F}_q$):
  Given an α-by-N matrix A, find a β-by-N matrix B where $\Delta_s(A,B) > k$.

Generalized Coding Problem

- For generators A, B, define:

$$\Delta_s(A,B) \equiv \min_{y\in C_A,\ y'\in C_B,\ y'\neq \mathbf{0}} \Delta(y,y')$$

  - *Note: Asymmetric*
  - *Note:* $\Delta_s(A,B) \leq$ *min-dist(*C_B*)*

- **“Span Distance Problem”** (over field $\mathbb{F}_q$): Given an α-by-N matrix A, find a β-by-N matrix B where $\Delta_s(A,B) > k$.

- Goal: Design code C_B with distance $> k$ to every codeword in C_A.

Filtered Secret sharing $\Longleftrightarrow$ Span Distance

- Filtered secret sharing (linear f) is a special case of the span distance problem:

$\exists$ (linear) f.s.s. solution f with $t = n - ck$

$\Longleftrightarrow$ (for all $c \geq 1$)

$\exists$ solution to the **span distance problem** with

- $A = V^{\perp}$, (A generates null space of V)
- $\beta = t$,
- Required distance = k.

- Parameter $c \geq 1$ for network coding application: amount of capacity given up.

Positive result

- Given any A, choose B randomly.

Positive result

- Given any A, choose B randomly.
- Analysis like random lin. codes on G-V bound:
  - Each vector in C_B has $\leq |C_A|\text{Vol}_q(k, N)/q^N$ prob. of having dist. $\leq k$ from C_A.
  - Union bound over q^β codewords C_B:

Random B has $\Delta_s(A, B) > k$ w/ prob.

$$\begin{aligned} &\geq 1 - q^\beta q^\alpha q^{-N}\text{Vol}_q(k, N) \\ &= 1 - q^{-ck}\text{Vol}_q(k, N) \end{aligned}$$

Positive result

- Given any A, choose B randomly.
- Analysis like random lin. codes on G-V bound:
  - Each vector in C_B has $\leq |C_A|\text{Vol}_q(k,N)/q^N$ prob. of having dist. $\leq k$ from C_A.
  - Union bound over q^β codewords C_B:

Random B has $\Delta_s(A,B) > k$ w/ prob.

$$\begin{aligned} &\geq 1 - q^\beta q^\alpha q^{-N}\text{Vol}_q(k,N) \\ &= 1 - q^{-ck}\text{Vol}_q(k,N) \end{aligned}$$

- In general, Pr > 0 if $q > N^{\Omega(\frac{1}{c-1})}$.
- For $k = \Theta(N)$, need only $q > 2^{\Omega(1/(c-1))}$.

Negative Result: Using the Covering Radius

- *Covering radius*(C_A) = $\max_{z \in \mathbb{F}_q^N} \Delta(z, C_A)$.

Negative Result: Using the Covering Radius

- *Covering radius*(C_A) = $\max_{z \in \mathbb{F}_q^N} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^N$ (let alone subspace C_B) has dist $> k$ from C_A.

Negative Result: Using the Covering Radius

- *Covering radius*(C_A) = $\max_{z \in \mathbb{F}_q^N} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^N$ (let alone subspace C_B) has dist $> k$ from C_A.
- Setting $c = 1$, using result of [Cohen, Frankl 85]:

Thm: $\forall\ \alpha, \beta$ s.t. $\left(\alpha = N - \frac{\log N}{\log q} - \frac{\log \mathsf{Vol}_q(k,N)}{\log q} + 2\log N + \log q + \log\ln q\right)$ and $\left(k + \beta < N - \alpha = \frac{\log N}{\log q} + \frac{\log \mathsf{Vol}_q(k,N)}{\log q} - 2\log N - \log q - \log\ln q\right)$, $\exists\ A$ s.t. $\nexists\ B$ where $\Delta_s(A, B) > k = N - \alpha - \beta$.

Negative Result: Using the Covering Radius

- *Covering radius*(C_A) = $\max_{z \in \mathbb{F}_q^N} \Delta(z, C_A)$.
- If C_A has covering radius $k \implies$ no $b \in \mathbb{F}_q^N$ (let alone subspace C_B) has dist $> k$ from C_A.
- Setting $c = 1$, using result of [Cohen, Frankl 85]:

Thm: $\forall\, \alpha, \beta$ s.t. $\left(\alpha = N - \frac{\log N}{\log q} - \frac{\log \mathsf{Vol}_q(k,N)}{\log q} + 2\log N + \log q + \log\ln q\right)$ and $\left(k + \beta < N - \alpha = \frac{\log N}{\log q} + \frac{\log \mathsf{Vol}_q(k,N)}{\log q} - 2\log N - \log q - \log\ln q\right)$, $\exists\, A$ s.t. $\nexists\, B$ where $\Delta_s(A, B) > k = N - \alpha - \beta$.

- $\exists$ reasonable settings of α, β, k, where $\exists A$ s.t. $q \geq N^{\Omega(\sqrt{k}/\log k)}$ if B exists. (Contrast to C/Y upper bound $\binom{N}{k}$.)

Conclusions

- Given a fixed linear network code, the problem of making it secure (using linear filtered secret sharing) is a generalized [classical] code design problem.
- To achieve security: trade-off between rate $(t = n - ck)$ and required link bandwidth $(\log q)$.
  - Sacrificing small amount of capacity allows large savings in required bandwidth.
- Secret sharing can be extended from [adversary gets $\leq k$ shares] ,to [adversary gets $\leq k$ *linear combinations* (from a given set) of *all* n shares] .

Future Work

- Better upper/lower bounds ($c = 1$, or in general).
- Consider (network topology, code design, security, robustness) simultaneously.
- Allow more power at nodes [Jain: random bits].
- Relax notion of security [Jain: computationally bounded adversary].
- Different adversaries, non-linear network codes, non-multicast?