Byzantine and pollution attacks

• Robustness against faulty/malicious components with arbitrary behavior, e.g.
  - dropping packets
  - misdirecting packets
  - sending spurious information

• Abstraction as Byzantine generals problem [LSP82]

• Byzantine robustness in networking [P88,MR97,KMM98,CL99]
Byzantine and pollution attacks

- Distributed randomized network coding can be extended to detect Byzantine behavior
  - Small computational and communication overhead
  - Small number of hash bits included with each packet, calculated as simple polynomial function of data
- Require only that a Byzantine attacker does not design and supply modified packets with complete knowledge of other nodes’ packets
- Main scheme:
  - Use a polynomial hash
  - An attacker without full knowledge of the traffic will have low probability of being able to match the hash
  - The hash can be used to detect an attack [HLKMEK04]
- One can further use such a hash to decode
Byzantine reliability

- Robustness against faulty/malicious components with arbitrary behavior, e.g.
  - dropping packets
  - misdirecting packets
  - sending spurious information
- Abstraction as Byzantine generals problem [LSP82]
- Byzantine robustness in networking [P88, MR97, KMM98, CL99]
Byzantine reliability

• Distributed randomized network coding can be extended to detect Byzantine behavior
  - Small computational and communication overhead
  - Small number of hash bits included with each packet, calculated as simple polynomial function of data
• Require only that a Byzantine attacker does not design and supply modified packets with complete knowledge of other nodes’ packets
• Let us use the approach of [HLKMEK04]
Byzantine reliability

- Data symbols $x_1, \ldots, x_\theta$
- Hash symbols $y_1, \ldots, y_\phi$

$$\pi(x_1, \ldots, x_k) = x_1^2 + \cdots + x_k^{k+1}$$

$$y_i = \pi(x_{(i-1)k+1}, \ldots, x_{ik}) \text{ for } i = 1, \ldots, \phi - 1$$

$$y_\phi = \pi(x_{(\phi-1)k+1}, \ldots, x_\theta)$$

$$k = \left\lfloor \frac{\theta}{\phi} \right\rfloor$$

Trades off overhead with probability of detection
If the receiver gets $s$ genuine packets, then the detection probability is at least

$$1 - \left(\frac{k+1}{q}\right)^s.$$ 

- With 2% overhead ($k = 50$), code length=7, $s = 5$, the detection probability is 98.9%.
- With 1% overhead ($k = 100$), code length=8, $s = 5$, the detection probability is 99.0%.