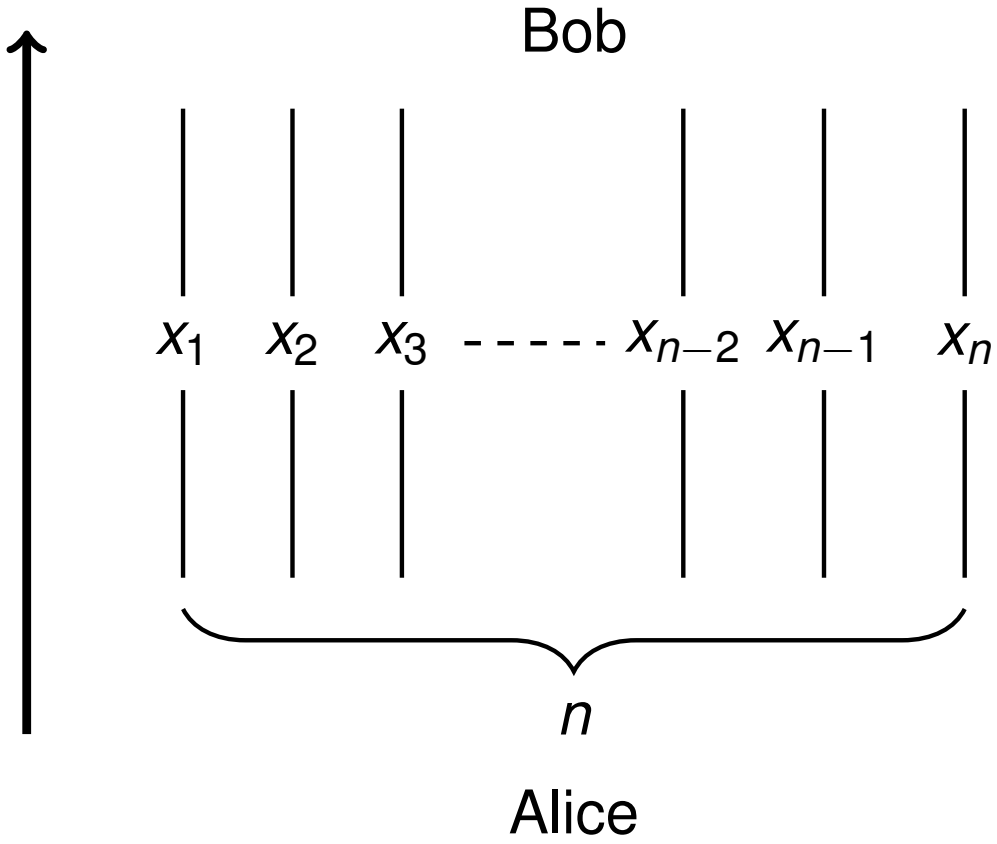
Perfectly Secure Message Transmission in Two Rounds

Gabriele Spini, **Gilles Zémor**

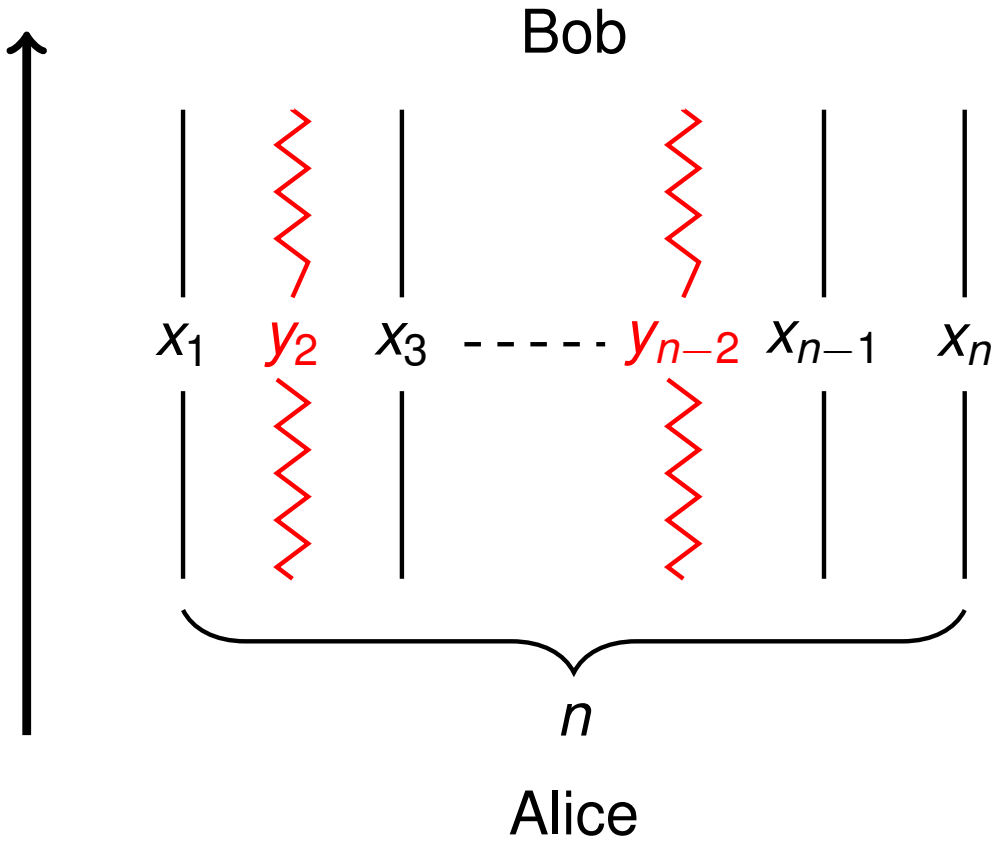
Bordeaux Mathematics Institute

October 2017, Descartes meeting

The problem: Alice wants to send private message to Bob through n parallel channels



The problem: Alice wants to send private message to Bob through n parallel channels



in presence of active adversary who controls t channels.

PSMT

Introduced by Dolev, Dwork, Waarts, Yung 1993.

Communication from Alice to Bob. Easy coding theory solution:

> Alice sends $\mathbf{x} = [x_1 \dots x_n]$ random word of t-error correcting code C such that the linear combination
>
> $$s = \langle \mathbf{h}, \mathbf{x} \rangle = h_1 x_1 + h_2 x_2 + \cdots + h_n x_n$$
>
> is the secret message.

$x_i, h_i, s \in \mathbb{F}_q$.

Wiretap II channel technique. Works as long as dim $C > t$, imposes $n \geq 3t + 1$.

What if $n < 3t + 1$?

PSMT in two rounds: n=2t+1

Allow two-way communication. First Bob sends message to Alice, then Alice to Bob.

Reliable and Private transmission of a secret from Alice to Bob is possible as long as $n \geq 2t + 1$.

Do so constructively and efficiently.

Efficiency:

$$\textit{Rate} = \frac{\text{total number of transmitted bits}}{\text{number of bits of secret message}}$$

$$\textit{Complexity} = \text{number of transmitted bits to convey 1-bit secret}$$

Results

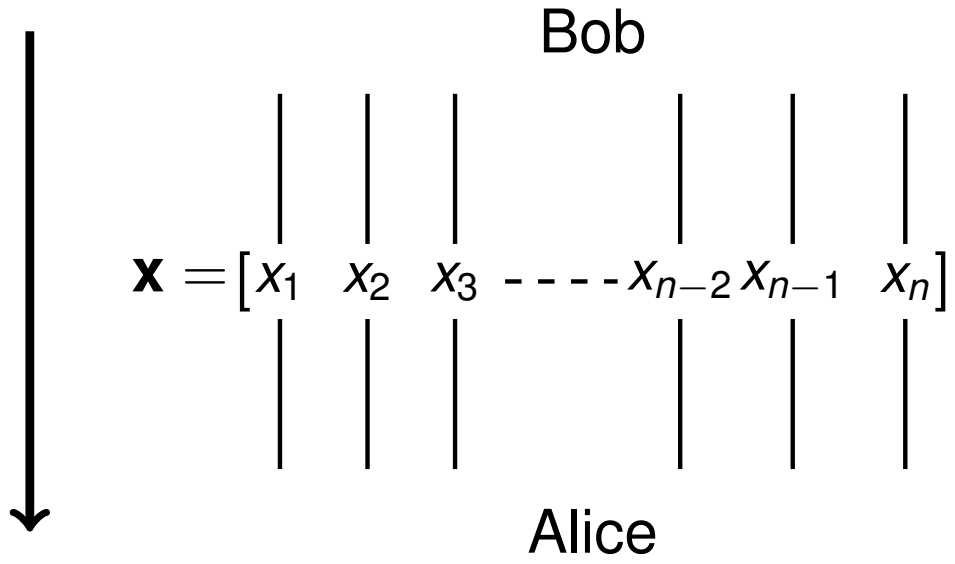
Previous work:

Sayeed and Abou-Amara 1996, Srinathan, Narayanan, and Rangan (Crypto 2004) Agarwal, Cramer, and de Haan (Crypto 2006) Kurosawa and Suzuki (Eurocrypt 2008).

Best previous protocol, Kurosawa and Suzuki: Rate = λn, Complexity = $\lambda n^3 \log n$.

This contribution: Complexity = $\lambda n^2 \log n$.
(Also improved λ for the rate)

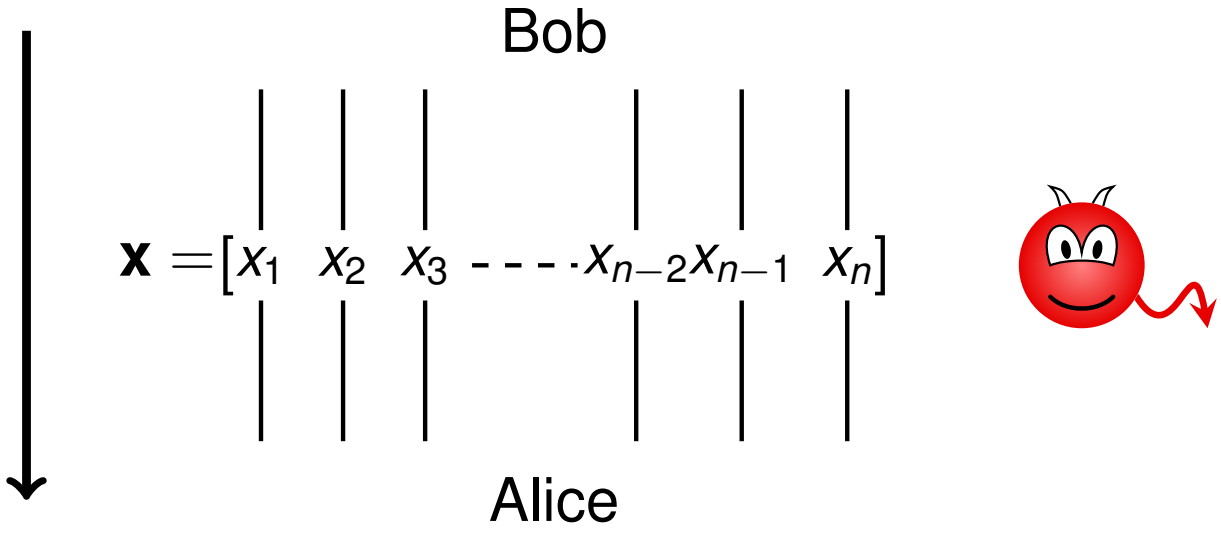
Results



Novel defining feature of protocol: *simplicity*.

Bob only sends $\mathbf{x}$ codeword of fixed MDS code.

Simplified scenario



Adversary is passive during first Bob $\to$ Alice phase.
Alice $\to$ Bob phase: Alice *broadcasts*

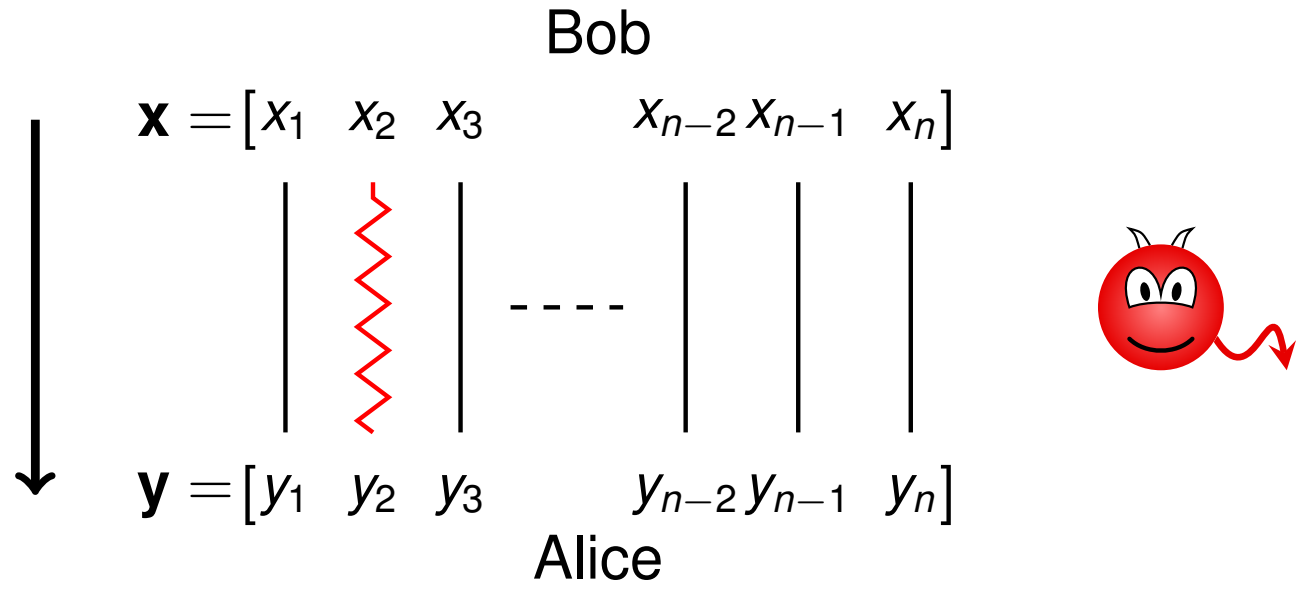
$$s + \langle \mathbf{h}, \mathbf{x} \rangle = s + h_1 x_1 + h_2 x_2 + \cdots + h_n x_n$$

(broadcast x: send $[x, x, \ldots, x]$)

$s \in \mathbb{F}_q$ is secret message. $\mathbf{x}$ is random codeword of C $[n = 2t + 1, t + 1, t + 1]$ MDS code.

Simplified scenario II

Adversary is active during Bob $\rightarrow$ Alice phase, but *not too much*, introduces at most $t/2$ errors.



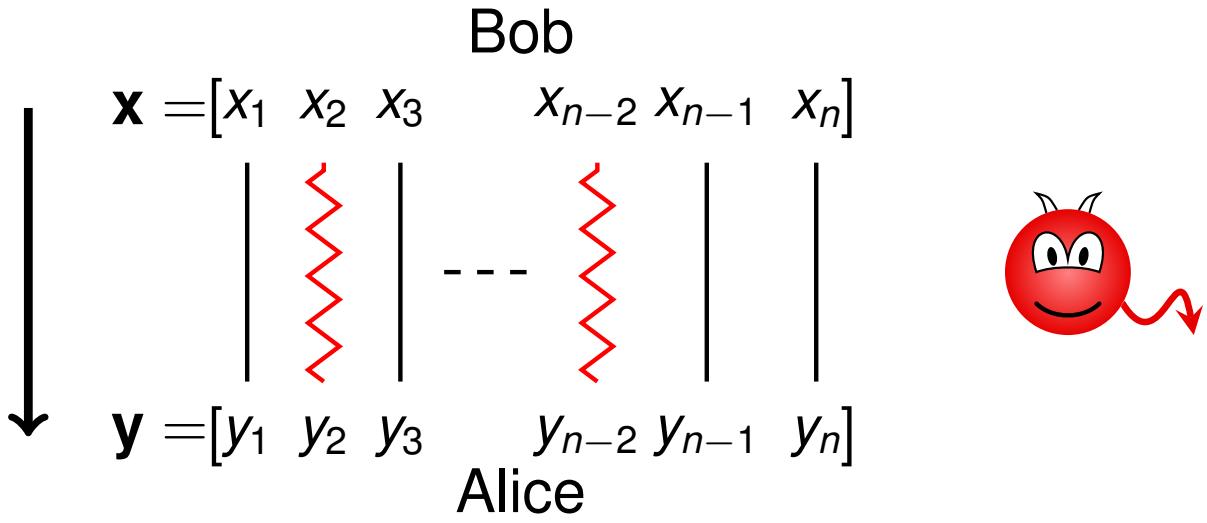
Alice broadcasts:

$$s + \langle \mathbf{h}, \mathbf{y} \rangle$$

$$\sigma(\mathbf{y}) = \mathbf{H}\mathbf{y}^T = \sigma(\mathbf{e})$$

Simplified Scenario III

Adversary is fully active during Bob $\rightarrow$ Alice phase, but a genie tells Bob what are the channels on which the adversary has introduced errors.

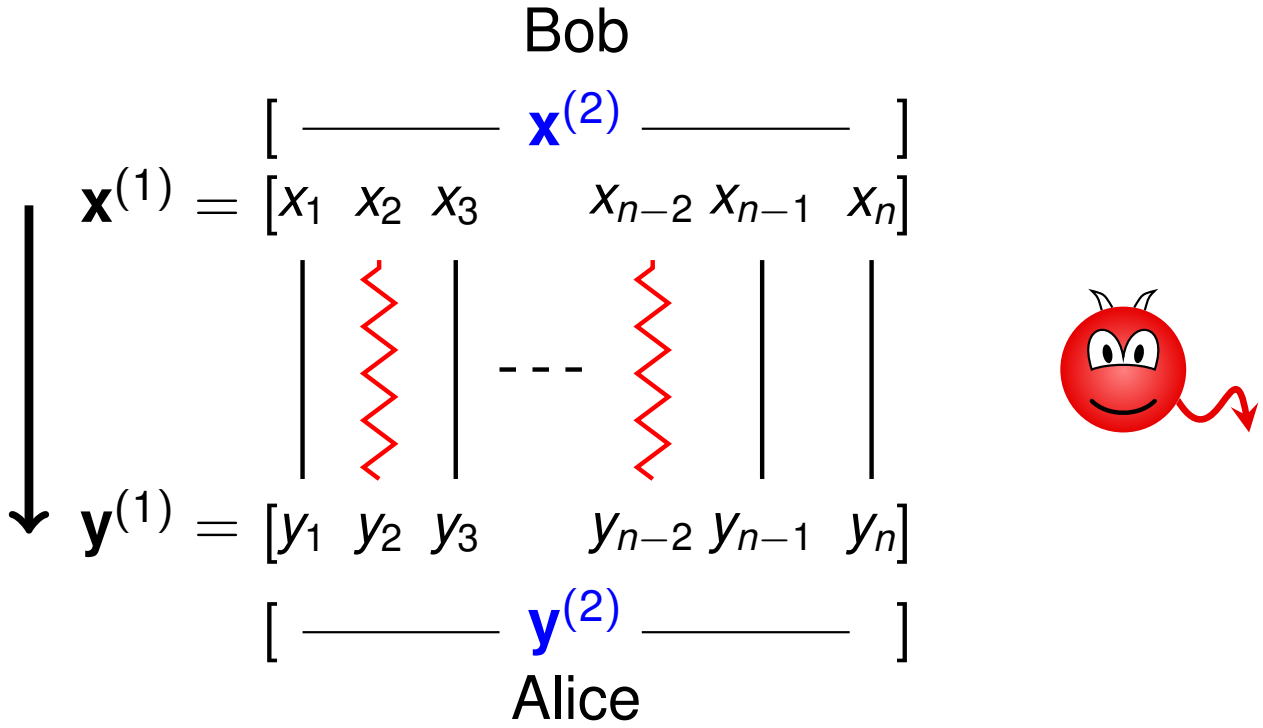


As before, Alice sends $s + \langle \mathbf{h}, \mathbf{y} \rangle, \quad \sigma(\mathbf{y}) = \mathbf{H}\mathbf{y}^T$.

Genie has transformed Bob's error decoding from syndrome problem into an *erasure* decoding from the syndrome problem. Code C can correct t erasures.

Almost complete scenario

Adversary corrupts every symbol of $\mathbf{x}^{(1)}$ on every one of the t channels it controls.



Alice broadcasts $s + \langle \mathbf{h}, \mathbf{y}^{(2)} \rangle$, $\sigma(\mathbf{y}^{(2)})$, and $\mathbf{y}^{(1)}$.

$\mathbf{y}^{(1)}$ reveals corrupted channels to Bob. *Alice* is the genie.

Complete scenario

Bob sends $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}$,
Alice receives $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(t)}, \mathbf{y}^{(t+1)}$.

Alice finds *proper* subset I of $\{1, 2, \ldots, t+1\}$ such that *every channel* used to corrupt *any* $\mathbf{x}^{(j)}$, $j \notin I$, was also used to corrupt some $\mathbf{x}^{(i)}$, $i \in I$.

Alice broadcasts all $\mathbf{y}^{(i)}$, $i \in I$. This reveals to Bob *all* channels used to corrupt *all* codewords $\mathbf{x}$. Alice does the genie's work.

Alice broadcasts (as before), for some $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$, $j \notin I$,

$$s + \langle \mathbf{h}, \mathbf{y} \rangle, \quad \sigma(\mathbf{y})$$

How does Alice find the set I ??

Finding the set I

E vector space of $\mathbb{F}_q^n$ generated by all errors introduced by adversary. Syndrome function

$$\sigma \ : E \to \mathbb{F}_q^{t+1}$$

is *injective* on E. So, $(\mathbf{e}_i)_{i\in I}$ basis of E iff $\sigma(\mathbf{e}_i)_{i\in I}$ basis of $\sigma(E)$.

Since for $\mathbf{y} = \mathbf{x} + \mathbf{e}$, $\sigma(\mathbf{y}) = \sigma(\mathbf{e})$,

Alice computes $\sigma(\mathbf{y}_1), \sigma(\mathbf{y}_2), \ldots, \sigma(\mathbf{y}_{t+1})$, finds a basis

$$(\sigma(\mathbf{y}^{(i)}))_{i\in I}$$

gives the required set of revealing vectors $(\mathbf{y}^{(i)})_{i\in I}$.

(Pseudo-basis of the set of received vectors $\mathbf{y}$).

Efficient transmission of pseudo-basis

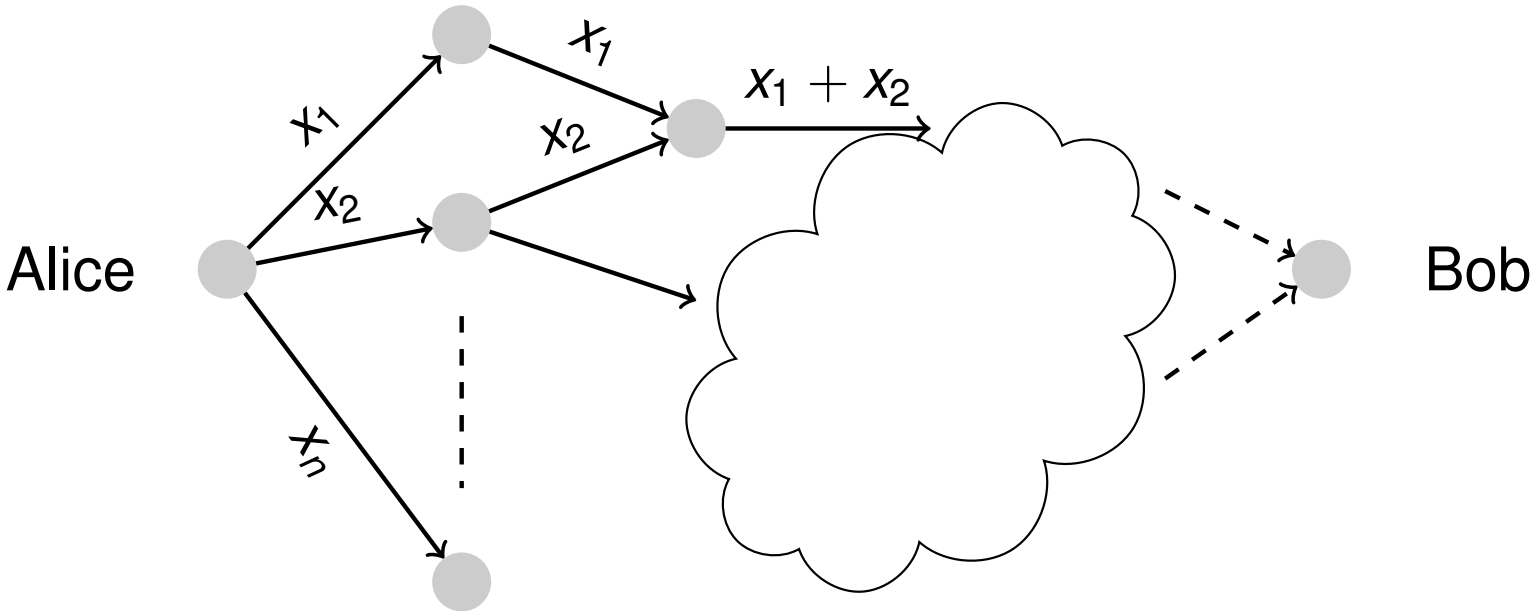
Broadcasting a symbol x as $[x, x, \ldots x,]$ costs n, broadcasting a vector costs n^2, broadcasting t vectors costs n^3.

Generalized broadcast:

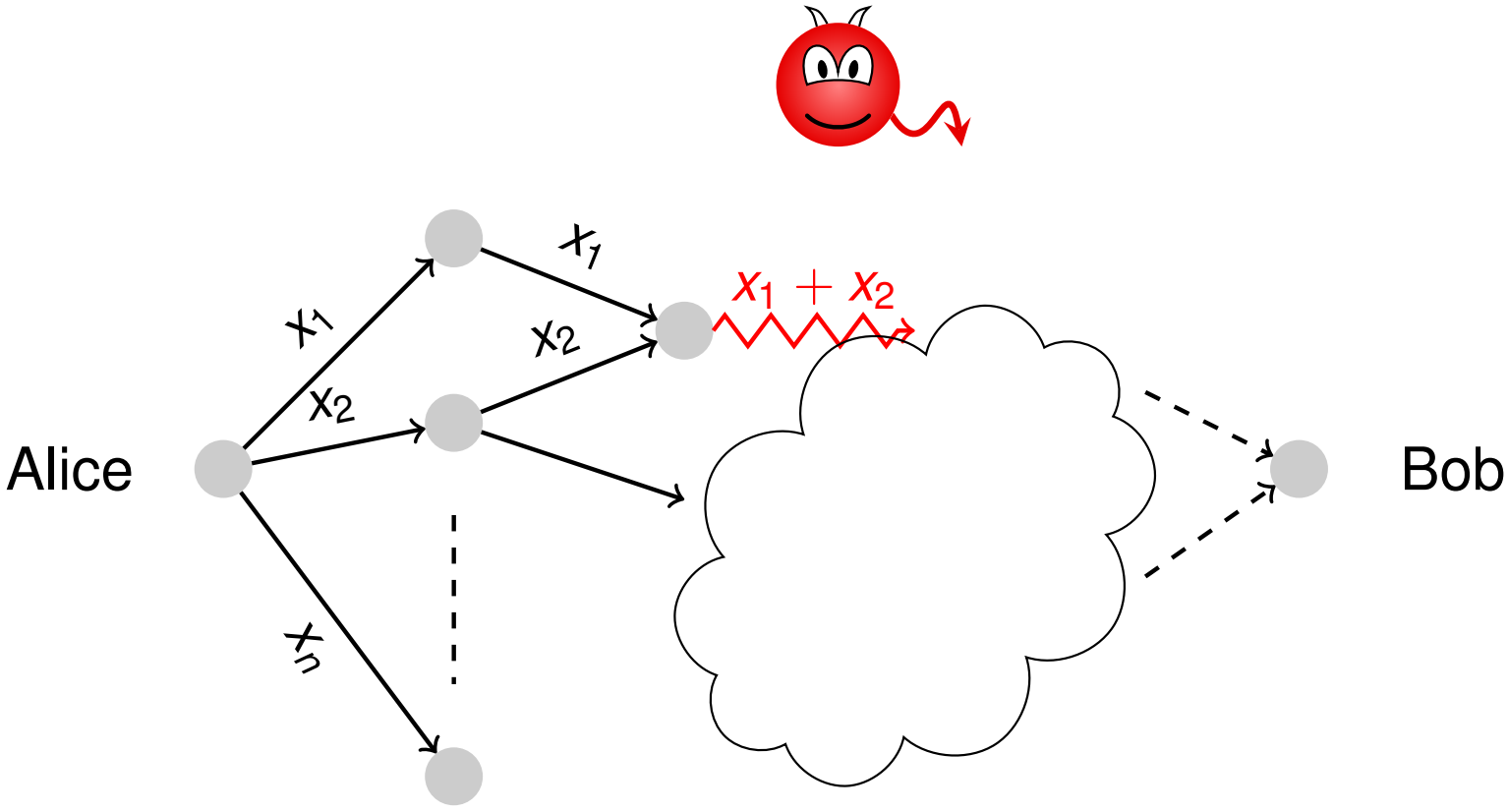
- $[x, x, \ldots x,]$ is $[n, 1, n]$ repetition code.
- After having sent first vector $\mathbf{y}$, at least one corrupted channel is revealed. So Bob needs to correct at worst $t - 1$ errors and 1 erasure. Use $[n, 2, n - 1]$ code.
- After having sent second vector $\mathbf{y}$ use $[n, 3, n - 2]$ code.
- And so on.

requires sending $n^2 \log n$ symbols overall.

Application to Network Coding



Application to Network Coding



Adversary intercepts t arbitrary linear forms in coordinates $x_1, \ldots, x_n$ of $\mathbf{x}$.

Present PSMT protocol adapts.