A Storage-efficient and Robust Private Information Retrieval Scheme allowing few servers

D. Augot, Françoise Levy-dit-Vehel, Abudllatif Shikfa

INRIA, ENSTA, Alcatel-Lucent

Outline

- LDC codes
- Application to PIR
- Particular case of Reed-Muller and Derivative codes
- A better reduction

Information Theoretic PIR

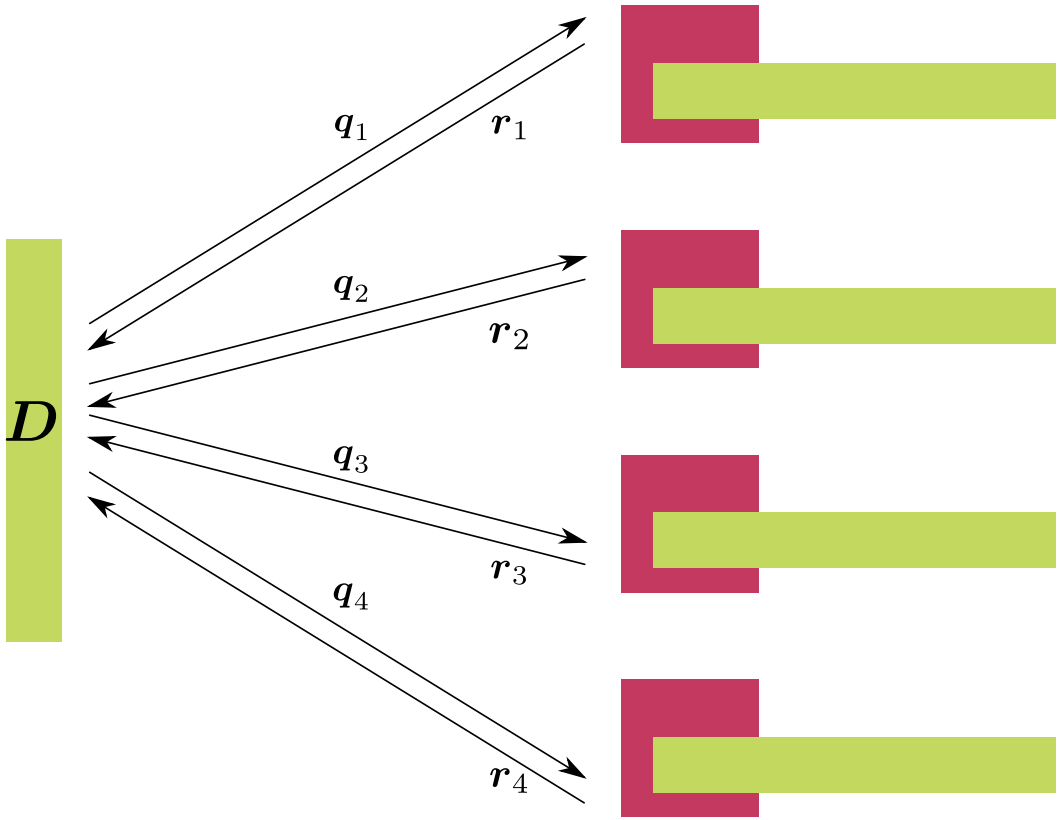
- User wants to retrieve $T[j]$ from a table T on a remote Server S
- Property: j and the returned $T[j]$ should be unknown to Server
- Scenario:
  - a centralized database of health records is kept by ObamaCare: doctors, nurses, etc, make queries about their patient, without revealing his identity
- *"information theoretic"*:

$$\Pr(j|\text{after the protocol is run}) = \Pr(j)$$

- With one server, we have a "Shannon-like"
  Theorem: The whole database has to be downloaded

Information theoretically secure PIR

With several servers:



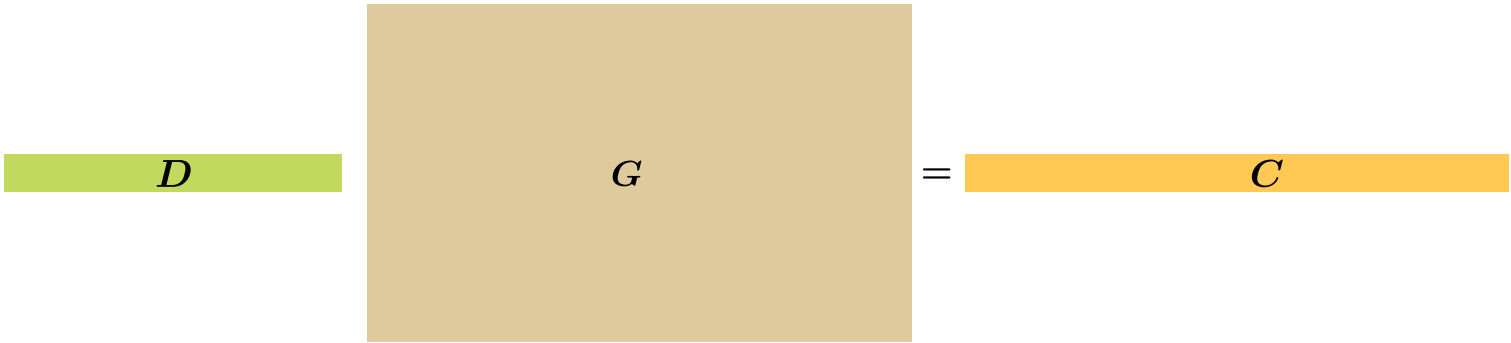
Can be achieved with *Locally Decodable Codes* (Katz-Trevisan00)

Coding

Definition (Code)

Given two alphabets Δ, Σ, a *code* is given by its encoding map $C : \Delta^k \to \Sigma^n$.

- The data (or message) D is transformed C into a (longer) codeword using a generating matrix (linear code)



- The rate is $\frac{k \log \Delta}{n \log \Sigma}$.

LDC (Locally Decodable Code) definition

A code $C : \Delta^k \to \Sigma^n$ is (ℓ, δ)-*locally decodable* if:

there exists a randomized decoding algorithm $\mathcal{A}^y(j)$ such that:

1. on input j, given *oracle access* to $y \in \Sigma^n$
2. $\mathcal{A}$ makes at most ℓ queries to y:

$$q_1, \ldots, q_\ell \in [1, n]$$

3. $\mathcal{A}$ receives $a_1, \ldots, a_\ell \in \Sigma$
4. compute $\tilde{x_j} = \mathcal{A}^y(j, a_1, \ldots, a_\ell) \in \Delta$
5. when

$$d(C(x), y) < \delta n$$

$$\Pr[\mathcal{A}^y(j) = x_j] \geq \frac{2}{3}$$

Furthermore, the code is *smooth* if each query q_j is uniform random.

LCC

Definition (Locally Correctable Code)

A code $C \subset \Sigma^n$ is (ℓ, δ)-*locally correctable* if
there exists a randomized decoding algorithm $\mathcal{A}$ such that:

1. given oracle access to $y \in \Sigma^n$, $\mathcal{A}$ makes at most ℓ queries to y.
2. on input i, output $\mathcal{A}^y(i)$
3. when $d(c, y) < \delta n$,

$$\Pr[\mathcal{A}^y(i) = c_i] \geq \frac{2}{3}$$

Theorem

A $\mathbb{F}_q$-linear LCC code $C \subset \mathbb{F}_q^n$ can be turned into an LDC code $C, Enc : \mathbb{F}_q^k \to \mathbb{F}_q^n$

Proof.

- Let $I \subset [1, n]$ be an information set of size $k = \dim C$.
- For $c \in C$ let $c_I \in \mathbb{F}_q^k$ denote the restriction of c to coordinates in I.
- Given a message $x \in \mathbb{F}_q^k$, we define $Enc(x)$ to be the unique element $c \in C$ such that $c_I = x$.
- Local correctability of C yields local decodability of C.

□

PIR

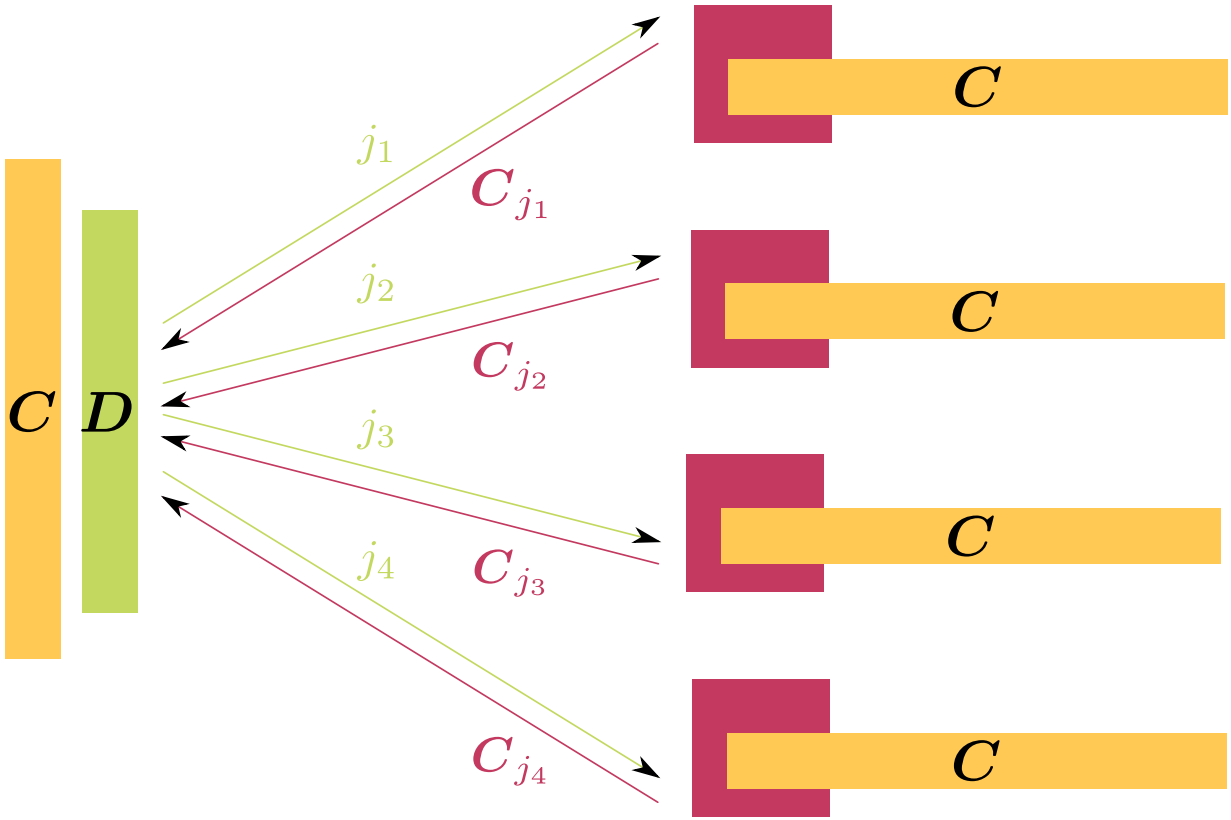
Definition (Private Information Retrieval (PIR))

An ℓ-server p-PIR protocol is a triple $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$ of algorithms as follows:

1. With a random string of bits s; User generates an ℓ-tuple of queries
$$(q_1, \ldots, q_\ell) = \mathcal{Q}(j, s)$$
2. For $1 \leq i \leq \ell$, User sends q_i to server S_i
3. Each S_i answers
$$a_i = \mathcal{A}(x, q_i)$$
to User;
4. User recovers $x_j = \mathcal{R}(a_1, \ldots, a_\ell, j, s)$ with probability p.

The protocol has *Privacy property* is $\Pr(j|q_j) = \Pr(j)$.

LDCs in PIR



The data D is encoded in a codeword C, which is stored on each server.

Reed-Muller codes

- We use the short-hand notation

$$\boldsymbol{X} = (X_1, \ldots, X_m) \qquad \boldsymbol{X}^{\boldsymbol{i}} = X_1^{i_1} \cdots X_m^{i_m}, \qquad \mathbb{F}_q[\boldsymbol{X}] = \mathbb{F}_q[X_1, \ldots, X_m]$$
$$\boldsymbol{i} = (i_1, \ldots, i_m) \in \mathbb{N}^m \qquad |\boldsymbol{i}| = i_1 + \cdots + i_m$$

- For $\boldsymbol{i}, \boldsymbol{j} \in \mathbb{N}^m$, $\boldsymbol{i} \gg \boldsymbol{j}$ means $\forall\ u,\ i_u \geq j_u$,
- the $\boldsymbol{i}$-th Hasse derivative of $F \in \mathbb{F}_q[\boldsymbol{X}]$, denoted by $\mathrm{Hasse}(F, \boldsymbol{i})$, is

$$\mathrm{Hasse}(F, \boldsymbol{i})(\boldsymbol{X}) = \sum_{\boldsymbol{j} \gg \boldsymbol{i}} f_{\boldsymbol{j}} \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{X}^{\boldsymbol{j}-\boldsymbol{i}} \quad \text{with} \quad \binom{\boldsymbol{j}}{\boldsymbol{i}} = \binom{j_1}{i_1} \cdots \binom{j_m}{i_m},$$

$$F(\boldsymbol{X} + \boldsymbol{Z}) = \sum_{\boldsymbol{j}} f_{\boldsymbol{j}} (\boldsymbol{X} + \boldsymbol{Z})^{\boldsymbol{j}} = \sum_{\boldsymbol{i}} \mathrm{Hasse}(F, \boldsymbol{i})(\boldsymbol{X}) \boldsymbol{Z}^{\boldsymbol{i}},$$

Restriction to a line

- Consider a vector $\boldsymbol{V} \in \mathbb{F}_q^m \setminus \{0\}$, and a base point $\boldsymbol{P}$,
- consider the restriction of F to the line $D = \{\boldsymbol{P} + t\boldsymbol{V} : \ t \in \mathbb{F}_q\}$, which is a univariate polynomial that we denote by

$$F_{\boldsymbol{P},\boldsymbol{V}}(T) = F(\boldsymbol{P} + T\boldsymbol{V}) \in \mathbb{F}_q[T]$$

- We have the following relations:

$$F_{\boldsymbol{P},\boldsymbol{V}}(T) = \sum_{\boldsymbol{j}} \mathrm{Hasse}(F, \boldsymbol{j})(\boldsymbol{P}) \boldsymbol{V}^{\boldsymbol{j}} T^{|\boldsymbol{j}|},$$

$$\mathrm{coeff}(F_{\boldsymbol{P},\boldsymbol{V}}, i) = \sum_{|\boldsymbol{j}|=i} \mathrm{Hasse}(F, \boldsymbol{j})(\boldsymbol{P}) \boldsymbol{V}^{\boldsymbol{j}},$$

$$\mathrm{Hasse}(F_{\boldsymbol{P},\boldsymbol{V}}, i)(\alpha) = \sum_{|\boldsymbol{j}|=i} \mathrm{Hasse}(F, \boldsymbol{j})(\boldsymbol{P} + \alpha\boldsymbol{V}) \boldsymbol{V}^{\boldsymbol{j}}, \quad \alpha \in \mathbb{F}_q$$

Reed-Muller codes

- We enumerate $\mathbb{F}_q^m = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\}$,
- $\mathbb{F}_q[\boldsymbol{X}]_d$ is the set of polynomials of degree $\leq d$, with $d < q$.
- We *encode* polynomials using the *evaluation map*

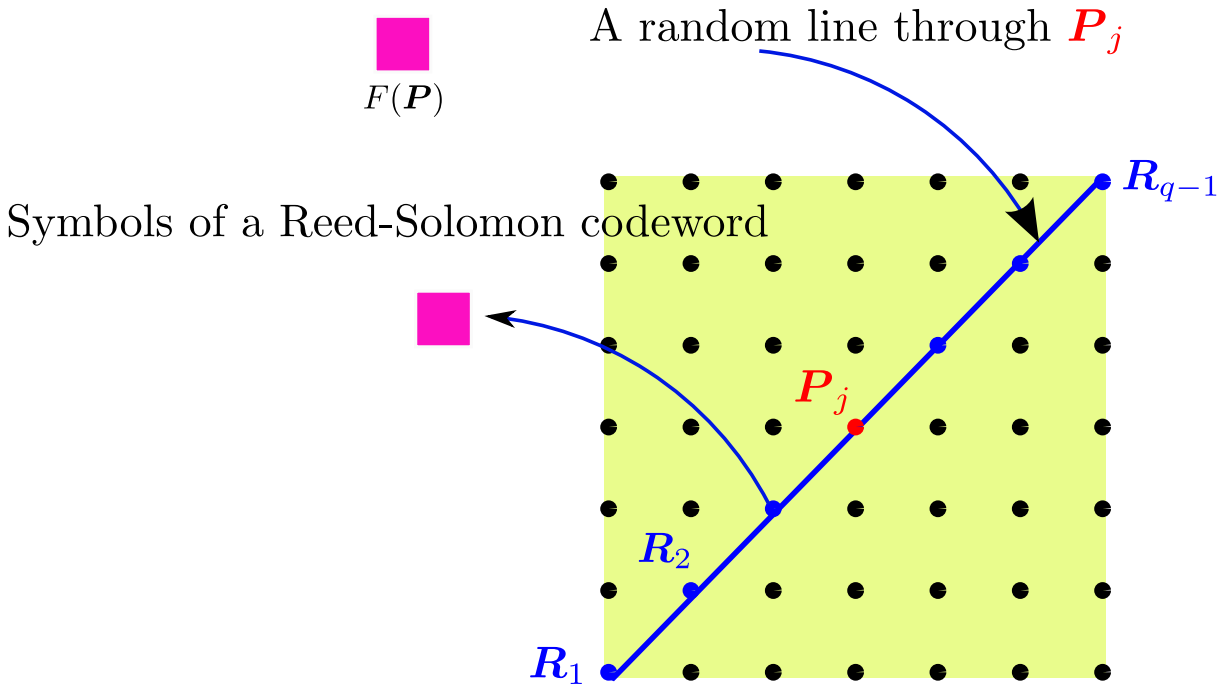
$$\begin{array}{rccc} \text{ev}: & \mathbb{F}_q[\boldsymbol{X}]_d & \to & \mathbb{F}_q^n \\ & F & \mapsto & (F(\boldsymbol{P}_1), \ldots, F(\boldsymbol{P}_n)) \end{array}$$

- The *d-th order Reed-Muller code* is

$$\text{RM}_d = \{\text{ev}(F) \mid F \in \mathbb{F}_q[\boldsymbol{X}]_d\}.$$

  with dimension $\binom{d+m}{m}$
- $c \in RM_d$ is indexed as $c = (c_{\boldsymbol{P}_1}, \ldots, c_{\boldsymbol{P}_n})$, where $c_i = c_{\boldsymbol{P}_i}$.

Local decoding of Reed-Muller codes



- $F(X, Y)$ restricted to a line $D(T)$ is a univariate polynomial
- for a given i, R_i is random, when the line is random
- two R_i's give the line $\Longrightarrow$ loss of incertitude (privacy) on $\boldsymbol{P}_j$

Local decoding of Reed-Muller codes

- Given *oracle access* to $y \approx c = \mathrm{ev}\, F$
- On input $\boldsymbol{P}_j$, pick a *random line* D of direction $\boldsymbol{V}$ passing through $\boldsymbol{P}_j$:

$$D(T) = \{\boldsymbol{P}_j + T \cdot \boldsymbol{V} \mid T \in \mathbb{F}_q\} = \{\boldsymbol{R}_0 = \boldsymbol{P}_j, \boldsymbol{R}_1 \ldots, \boldsymbol{R}_{q-1}\} \subset \mathbb{F}_q^m.$$

- $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{q-1}$ are sent as queries, and the decoding algorithm receives

$$\left(y_{\boldsymbol{R}_1}, \ldots, y_{\boldsymbol{R}_{q-1}}\right) \in \mathbb{F}_q^{q-1}.$$

- In case of no errors, $\left(y_{\boldsymbol{R}_1}, \ldots, y_{\boldsymbol{R}_{q-1}}\right) = \left(c_{\boldsymbol{R}_1}, \ldots, c_{\boldsymbol{R}_{q-1}}\right)$, and

$$c_{\boldsymbol{R}_u} = F(\boldsymbol{P}_j + \alpha_u \cdot \boldsymbol{V}) = F_{\boldsymbol{P},\boldsymbol{V}}(\alpha_u), \quad \alpha_u \neq 0$$

  where $F_{\boldsymbol{P},\boldsymbol{V}} = F(\boldsymbol{P} + T \cdot \boldsymbol{V}) \in \mathbb{F}_q[T]$ is the restriction of F to D.
- $F_{\boldsymbol{P},\boldsymbol{V}}$ is found with Lagrange interpolation, and $c_{\boldsymbol{P}_j} = F_{\boldsymbol{P},\boldsymbol{V}}(0)$.

Local decoding of Reed-Muller codes, in presence of errors

- Given *oracle access* to $y \approx c = \mathrm{ev}\, F$
- On input $\boldsymbol{P}_j$, pick a *random line* D of direction $\boldsymbol{V}$ passing through $\boldsymbol{P}_j$:

$$D(T) = \{\boldsymbol{P}_j + T \cdot \boldsymbol{V} \mid Tx \in \mathbb{F}_q\} = \{\boldsymbol{R}_0 = \boldsymbol{P}_j, \boldsymbol{R}_1 \ldots, \boldsymbol{R}_{q-1}\} \subset \mathbb{F}_q^m.$$

- $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{q-1}$ are sent as queries, and the decoding algorithm receives

$$\left(y_{\boldsymbol{R}_1}, \ldots, y_{\boldsymbol{R}_{q-1}}\right) \in \mathbb{F}_q^{q-1}.$$

- $\left(y_{\boldsymbol{R}_1}, \ldots, y_{\boldsymbol{R}_{q-1}}\right) \approx \left(c_{\boldsymbol{R}_1}, \ldots, c_{\boldsymbol{R}_{q-1}}\right) = \mathrm{ev}_{\mathrm{RS}}(F(\boldsymbol{P} + T \cdot \boldsymbol{V})) = \mathrm{ev}_{\mathrm{RS}}(F_{\boldsymbol{P},\boldsymbol{V}})$
- One can recover $F_{\boldsymbol{P}_j,\boldsymbol{V}}$, using a Reed-Solomon decoding algorithm, and compute $c_{\boldsymbol{P}_j} = F_{\boldsymbol{P}_j,\boldsymbol{V}}(0)$.

Multiciplity codes (Kopparty-Saraf-Yekhanin2011)

- Let $s \in \mathbb{N}$, and $\sigma = \binom{m+s-1}{m}$, we build the evaluation at a point $\boldsymbol{P}$:

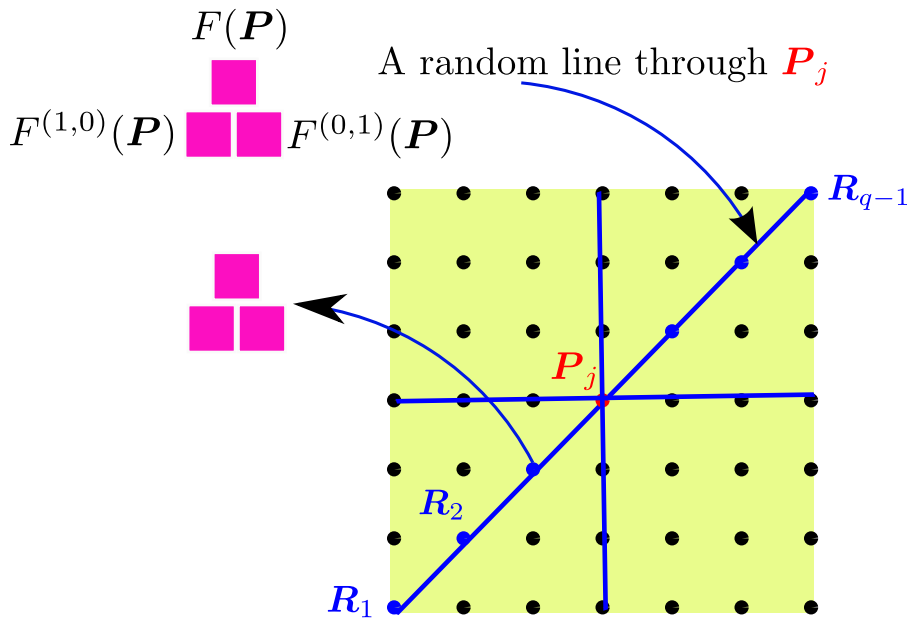
$$\begin{array}{rccl} \mathrm{ev}_{\boldsymbol{P}}^{s} : & \mathbb{F}_q[\boldsymbol{X}] & \to & \mathbb{F}_q^{\sigma} \\ & F & \mapsto & (\mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}))_{|\boldsymbol{v}|<s} \end{array}$$

- Consider $\mathbb{F}_q[\boldsymbol{X}]_d$, with $d < s(q-1)$, the corresponding code is

$$\mathrm{Mult}_d^s = \left\{ \left(\mathrm{ev}_{\boldsymbol{P}_i}^s(F)\right)_{i=1,\ldots,n} \mid F \in \mathbb{F}_q[\boldsymbol{X}]_d \right\}.$$

- It is a code $\mathrm{Mult}_d^s : \Delta^k \to \Sigma^n$, with $\Delta = \mathbb{F}_q$, and $\Sigma = \mathbb{F}_q^{\sigma}$.
- The code Mult_d^s is a $\mathbb{F}_q$-linear with dimension $k = \binom{m+d}{d}$.
- rate $R = \binom{m+d}{m} / \left(\binom{m+s-1}{m} \cdot q^m \right)$
- minimum distance $q^m - \frac{d}{s} q^{m-1}$ (Generalized Schwartz-Zippel).

Local Decoding



- Two variables, first order derivatives $= 1$, $\sigma = 3$

$$\mathrm{ev}_{\boldsymbol{P}}(F) = (F(\boldsymbol{P}), F^{(1,0)}(\boldsymbol{P}), F^{(0,1)}(\boldsymbol{P})$$

- three random lines are needed, locality is $(q-1)\sigma$

- Given $y = (y_1, \ldots, y_n)$, a noisy version of $c = \mathrm{ev}^s(F) \in \mathrm{Mult}_d$.
- Input $j \in [n]$
- Pick distinct σ non zero distinct random vectors $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_\sigma$
- For $i = 1$ to σ:
  - Consider the line

$$\{\boldsymbol{P}_j + 0 \cdot \boldsymbol{U}_i, \boldsymbol{P}_j + \alpha_1 \cdot \boldsymbol{U}_i, \ldots, \boldsymbol{P}_j + \alpha_{q-1} \cdot \boldsymbol{U}_i\} = \{\boldsymbol{R}_{i,0}, \ldots, \boldsymbol{R}_{i,q-1}\}$$

  - Send $\boldsymbol{R}_{i,1}, \ldots, \boldsymbol{R}_{i,q-1}$, as queries,
  - Receive the answers: $(y_{\boldsymbol{R}_{i,b}})_{\boldsymbol{v}} = \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{R}_{i,b})$,
  - Compute for each point, and for each order $0 \leq e < s$

$$\mathrm{Hasse}(F_{\boldsymbol{P}_j, \boldsymbol{U}_i}, e)(\alpha_b) = \sum_{|\boldsymbol{v}|=e} \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{R}_{i,b}) \boldsymbol{U}_i^{\boldsymbol{v}}$$

  - Recover $F_{\boldsymbol{P}_j, \boldsymbol{U}_i}$ by Hermite interpolation (no error).

- Solve for the indeterminates $\mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j)$, $|\boldsymbol{v}| < s$, the system:

$$\mathrm{coeff}(F_{\boldsymbol{P}_j, \boldsymbol{U}_i}, t) = \sum_{|\boldsymbol{v}|=t} \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j) \boldsymbol{U}_i^{\boldsymbol{v}}. \quad \begin{cases} t = 0, \ldots, s-1, \\ i = 1, \ldots \sigma \end{cases}$$

- Return $\{\mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j),\ |\boldsymbol{v}| < s\} = \mathrm{ev}^s_{\boldsymbol{P}_j}(F)$.

- Given $y = (y_1, \ldots, y_n)$, a noisy version of $c = \mathrm{ev}^s(F) \in \mathrm{Mult}_d$.
- Input $j \in [n]$
- Pick distinct σ non zero distinct random vectors $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_\sigma$
- For $i = 1$ to σ:
  - Consider the line

$$\{\boldsymbol{P}_j + 0 \cdot \boldsymbol{U}_i, \boldsymbol{P}_j + \alpha_1 \cdot \boldsymbol{U}_i, \ldots, \boldsymbol{P}_j + \alpha_{q-1} \cdot \boldsymbol{U}_i\} = \{\boldsymbol{R}_{i,0}, \ldots, \boldsymbol{R}_{i,q-1}\}$$

  - Send $\boldsymbol{R}_{i,1}, \ldots, \boldsymbol{R}_{i,q-1}$, as queries,
  - Receive the answers: $(y_{\boldsymbol{R}_{i,b}})_{\boldsymbol{v}} = \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{R}_{i,b})$,
  - Compute for each point, and for each order $0 \leq e < s$

$$\mathrm{Hasse}(F_{\boldsymbol{P}_j, \boldsymbol{U}_i}, e)(\alpha_b) = \sum_{|\boldsymbol{v}|=e} \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{R}_{i,b}) \boldsymbol{U}_i^{\boldsymbol{v}}$$

  - Recover $F_{\boldsymbol{P}_j, \boldsymbol{U}_i}$ by Hermite interpolation (no error).
    Or decoding of multiplity Reed-Solomon code
- Solve for the indeterminates $\mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j)$, $|\boldsymbol{v}| < s$, the system:

$$\mathrm{coeff}(F_{\boldsymbol{P}_j, \boldsymbol{U}_i}, t) = \sum_{|\boldsymbol{v}|=t} \mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j) \boldsymbol{U}_i^{\boldsymbol{v}}. \quad \begin{cases} t = 0, \ldots, s-1, \\ i = 1, \ldots \sigma \end{cases}$$

- Return $\{\mathrm{Hasse}(F, \boldsymbol{v})(\boldsymbol{P}_j),\ |\boldsymbol{v}| < s\} = \mathrm{ev}^s_{\boldsymbol{P}_j}(F)$.

- For ß > 0, $\Sigma = \mathbb{F}_q^s$, and MRS $= \{c = \mathrm{ev}^s(F) \mid \ F \in \mathbb{F}_q[X]_d\}$.
- Decoding t errors is, for $y \in \Sigma^n$, find polynomials $F \in \mathbb{F}_q[X]_d$ s.t.

$$d_\Sigma(\mathrm{ev}^s(F), y) \leq t$$

- Find $N,\ E \in \mathbb{F}_q[X]$ of degrees $(sn+d)/2$ and $(sn-d)/2$, s.t. for $i = 1, \ldots, n$

$$\begin{cases} N(\alpha_i) &= E(\alpha_i) \cdot y_{i,0} \\ &\vdots \\ \mathrm{Hasse}(N, s-1)(\alpha_i) &= \sum_{j=0}^{s-1} \mathrm{Hasse}(E, j)(\alpha_i) \cdot y_{i,s-1-j} \end{cases}$$

- A non-zero solution (N, E) always exists, F is recovered as N/E.
- With $t = (n - d/s)/2$, F will satisfy $N - EF = 0$
  - for any α_u s.t. that $\mathrm{ev}^s_{\alpha_u}(F) = y_u$, $N - EF$ has a zero of multiplicity s at α_u:

$$\sum_{i=1,\ldots,n} \mathrm{mult}(N - EF, \alpha_i) > (n-t)s = (sn+d)/2.$$

  - $\deg(N - EF) \leq (sn+d)/2$.

Our contribution: partitioning

- a $\mathbb{F}_q$-linear hyperplane of $\mathbb{F}_q^m$ H is the kernel of

$$f_H : (x_1, \ldots, x_m) \mapsto h_1 x_1 + \cdots h_m x_m$$

- $\mathbb{F}_q^m$ can be *split* as the *disjoint union of affine hyperplanes*

$$\mathbb{F}_q^m = H_0 \cup H_1 \cup \cdots \cup H_{q-1}, \quad H_i = f_H^{-1}(\alpha_i)$$

- For example $H = \{x_m = 0\}$, we have

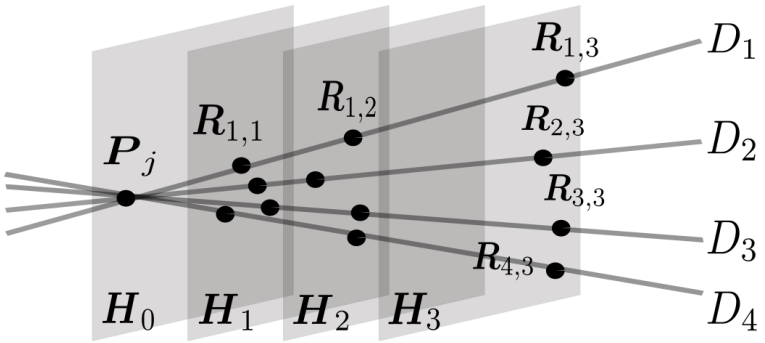
$$\mathbb{F}_q^m = H_0 \cup H_1 \cup \ldots H_{q-1}, \quad H_i = \{x_m = \alpha_i\}$$

- Up to a permutation, we write codewords $c = \left(c_{H_0} | \cdots | c_{H_{q-1}}\right)$, where

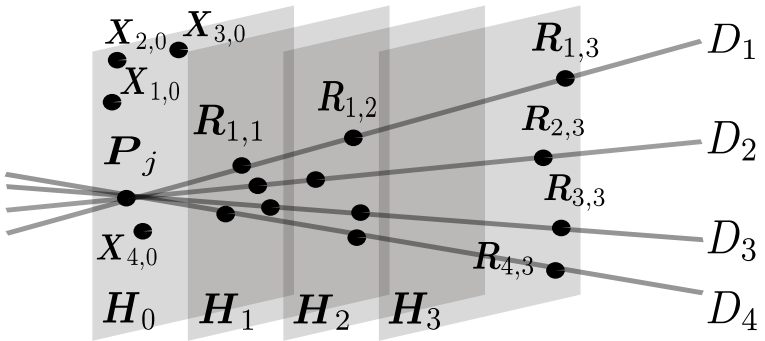
$$c_{H_i} = (\mathrm{ev}_{\boldsymbol{P}}(f))_{\boldsymbol{P} \in H_i}, \ i = 0 \ldots, q-1.$$

Local decoding with transversal lines

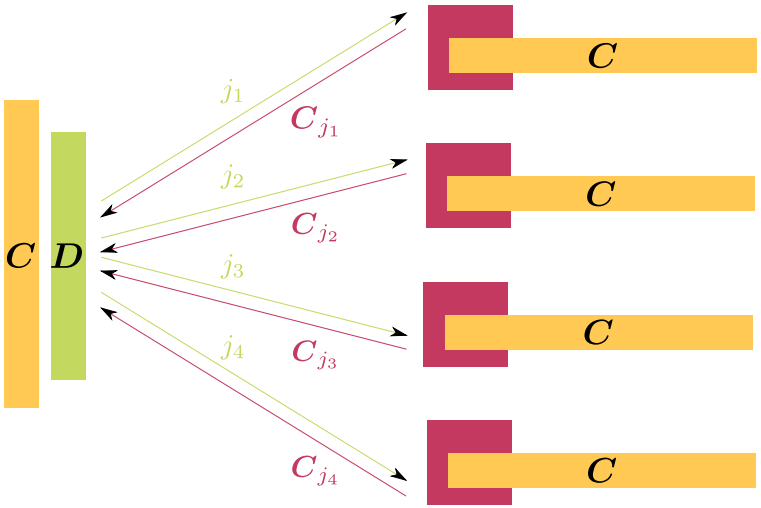
- Consider a line, *transversal* to the hyperplanes,
- it is given by $\boldsymbol{U} \in \mathbb{F}_q^m$ s.t. $f_H(\boldsymbol{U}) \neq 0$: $D = \{\boldsymbol{P} + t \cdot \boldsymbol{U} \mid t \in \mathbb{F}_q\}$



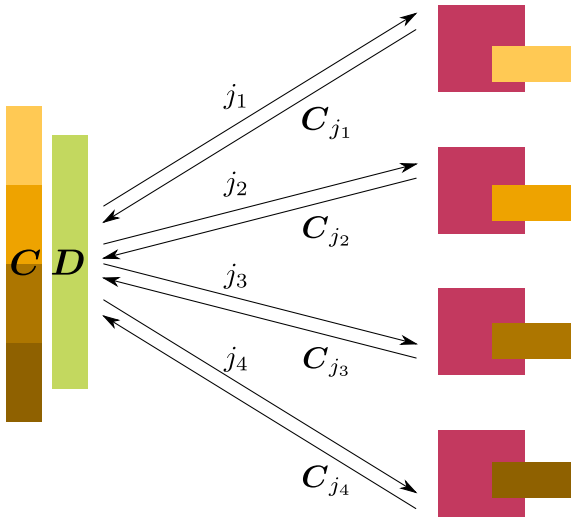
- We obfuscate with random $\boldsymbol{X}_i$'s the Server which holds $\boldsymbol{P}_j$



Our new LDC to PIR



Previous



Now

- Applies to Reed-Muller codes
- We think it is not generic.

Table $q = 16$

- Our PIR locality ℓ' is q, instead of $(q-1)\sigma = \binom{m+s-1}{m}$
- Our global storage overhead is $1/R$, instead of ℓ/R

Parameters				Locality		Stor. overhead		Comm. complexity	
m	s	d	k	♯ queries	♯ servers	std	ours	std	ours
2	1	14	120	15	16	32	2.1	180	128
2	2	29	465	45	16	25	1.7	900	768
2	3	44	1035	90	16	22	1.5	2880	2688
2	4	59	1830	150	16	21	1.4	7200	7040
2	5	74	2850	225	16	20	1.3	15300	15360
2	6	89	4095	315	16	20	1.3	28980	29568
3	1	14	680	15	16	90	6.0	240	192
3	2	29	4960	60	16	50	3.3	1680	1536
3	3	44	16215	150	16	38	2.5	7800	7680
3	4	59	37820	300	16	32	2.2	27600	28160
3	5	74	73150	525	16	29	2.0	79800	82880
3	6	89	125580	840	16	27	1.8	198240	207872
4	1	14	3060	15	16	320	21	300	256
4	2	29	40920	75	16	120	8.0	2700	2560
4	3	44	194580	225	16	76	5.1	17100	17280
4	4	59	595665	525	16	58	3.9	81900	85120
4	5	74	1426425	1050	16	48	3.2	310800	327040
4	6	89	2919735	1890	16	42	2.8	982800	1040256

Table $q = 256$

- Our PIR locality ℓ' is q, instead of $(q-1)\sigma = \binom{m+s-1}{m}$
- Our global storage overhead is $1/R$, instead of ℓ/R

Parameters				Locality		Stor. overhead		Comm. complexity	
m	s	d	k	♯ queries	♯ servers	std	ours	std	ours
2	1	254	32640	255	256	510	2.0	6120	4096
2	2	509	130305	765	256	380	1.5	30600	24576
2	3	764	292995	1530	256	340	1.3	97920	86016
2	4	1019	520710	2550	256	320	1.3	244800	225280
2	5	1274	813450	3825	256	310	1.2	520200	491520
2	6	1529	1171215	5355	256	300	1.2	985320	946176
3	1	254	2796160	255	256	1500	6.0	8160	6144
3	2	509	22238720	1020	256	770	3.0	57120	49152
3	3	764	74909055	2550	256	570	2.2	265200	245760
3	4	1019	177388540	5100	256	480	1.9	938400	901120
3	5	1274	346258550	8925	256	430	1.7	2713200	2652160
3	6	1529	598100460	14280	256	400	1.6	6740160	6651904
4	1	254	180352320	255	256	6100	24	10200	8192
4	2	509	2852115840	1275	256	1900	7.5	91800	81920
4	3	764	14382538560	3825	256	1100	4.5	581400	552960
4	4	1019	45367119105	8925	256	840	3.3	2784600	2723840
4	5	1274	110629606725	17850	256	690	2.7	10567200	10465280
4	6	1529	229222001295	32130	256	600	2.4	33415200	33288192

Size of q

- Being *information theoretically secure*, our protocol can be run any number of times. This implies that q can be chosen freely, and does not need to have a special relationship with the natural date.
- Imagine a database of *900 000 IPV6 adresses*, each with 128 bits=16 bytes. It needs $90\,000 \cdot 16 = 1,440,000$ bytes of storage.
- $q = 256 = 2^8$, we need a $\mathbb{F}_q$-dimension at least $1,440,000$
  - We find a code of $\mathbb{F}_q$-dimension $2796160 \sim 2,7 \cdot 10^6$, and rate $1/6$.
  - The LDC-locality is 255, and its PIR-locality is 256.
  - The communication cost is 6,144 bits.
- $q_0 = 2^4 = 16$, we need a code of $\mathbb{F}_{q_0}$-dimension $2 \cdot 1\,440\,000$
  - We find a code of $\mathbb{F}_{q_0}$-dimension $2919735 \sim 2.9 \cdot 10^6$, and rate $1/2.8$.
  - Its LDC-locality is 1890 while its PIR-locality is 16.
  - But the communication cost is now 1,040,256 bits.

Results

- The standard reduction has an overhead of $\ell \cdot 1/R$. *Ours has only* $1/R$, which is the natural overhead of the code.
  - k/Rq symbols are required per server. In particular, when $R \geq 1/q$, each server stores less than k symbols (size of original database) !
- The locality is *only* q, opposed to $q\sigma = q\binom{m+s-1}{s}$
  - notion a PIR-locality different from LDC-locality
- The communication complexity stays $\approx$ the same.
- A lying server *"lies always on the same hyperplane"*:
  - We can decode if the local word has $t = \lfloor 1/2(q - 1 - d/s) \rfloor$ errors
  - Our protocol is a t-Byzantine robust
- Not at all robust to *collusions*: only two colluding servers find the line
  - incertitude drops from $1/q^m$ to $1/q$

Perspectives and open problems

- To ensure collusion resistance, replace lines par curves of higher degree
- Changing the hyperplane $H_m = \{x_m = 0\}$ provides (one-time) encryption (original Alcatel-Lucent idea)
- Not the same alphabet for message $\mathbb{F}_q$ and codewords $\mathbb{F}_q^\sigma$.

  The standard LCC to LDC reduction does not apply.
  Koppart et al. use concatenation.
  Notion of interpolating set $\neq$ information set

- Encoding is sloooow. Need to use fast arithmetic.
- Timings for decoding are very good (because the locality is very small), can be done on a smartphone.
- Locally decode with 2-dimensional planes instead on line could enable larger d and thus higher reates (Cuong's idea)
- Use the Generalized Reed-Muller codes (from coding theory), and study their locality. This could enable higher d and higher rate.

LCC

Definition (Locally Correctable Code)

A code $C : \Delta^k \to \Sigma^n$ is (ℓ, δ)-*locally correctable* if
there exists a randomized decoding algorithm $\mathcal{A}$ such that:

1. given oracle access to $y \in \Sigma^n$, $\mathcal{A}$ makes at most ℓ queries to y.
2. on input i, output $\mathcal{A}^y(i)$
3. when $d(C(x), y) < \delta n$,

$$\Pr[\mathcal{A}^y(i) = C(x)_i] \geq \frac{2}{3}$$

In the case of $\Delta = \Sigma = \mathbb{F}_q$, and the code being linear, LCC $\implies$ LDC (Yekahnin 2010).