Algebraic Geometry Codes

Background on Coding Theory

Background on Coding Theory

sender

receiver

m

$m' = m + e$

where e is some low Hamming weight noise term

Background on Coding Theory

sender

receiver

m

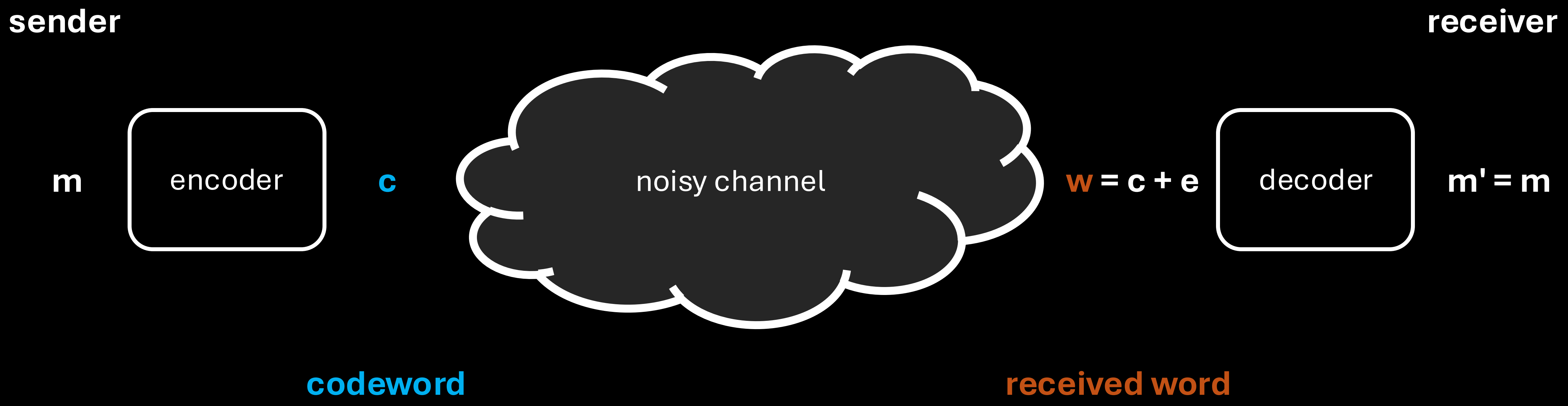
m' = m + e

where e is some low Hamming weight noise term

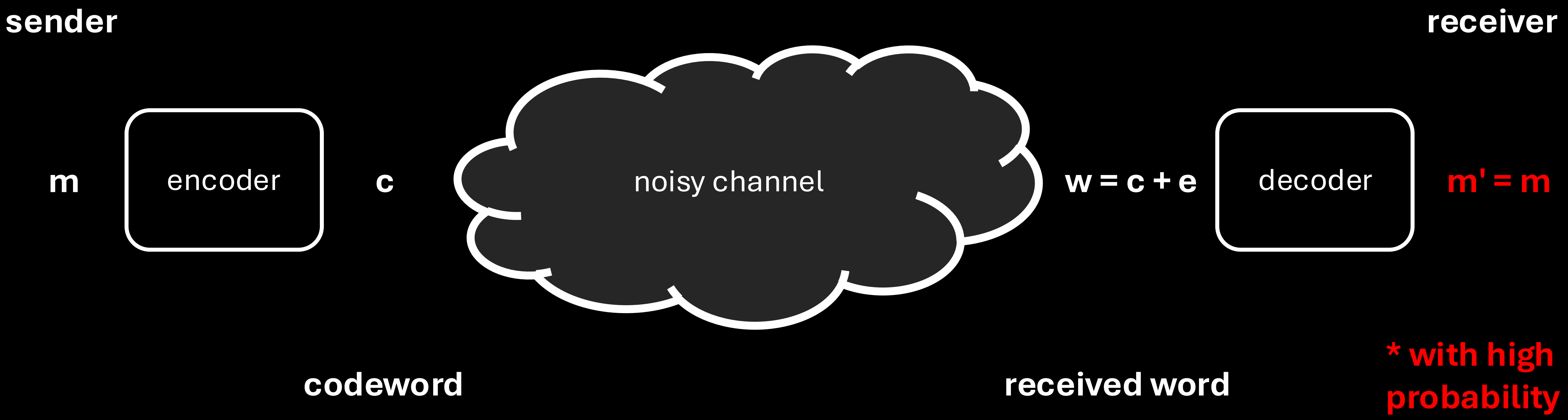
we don't want noise

Background on Coding Theory

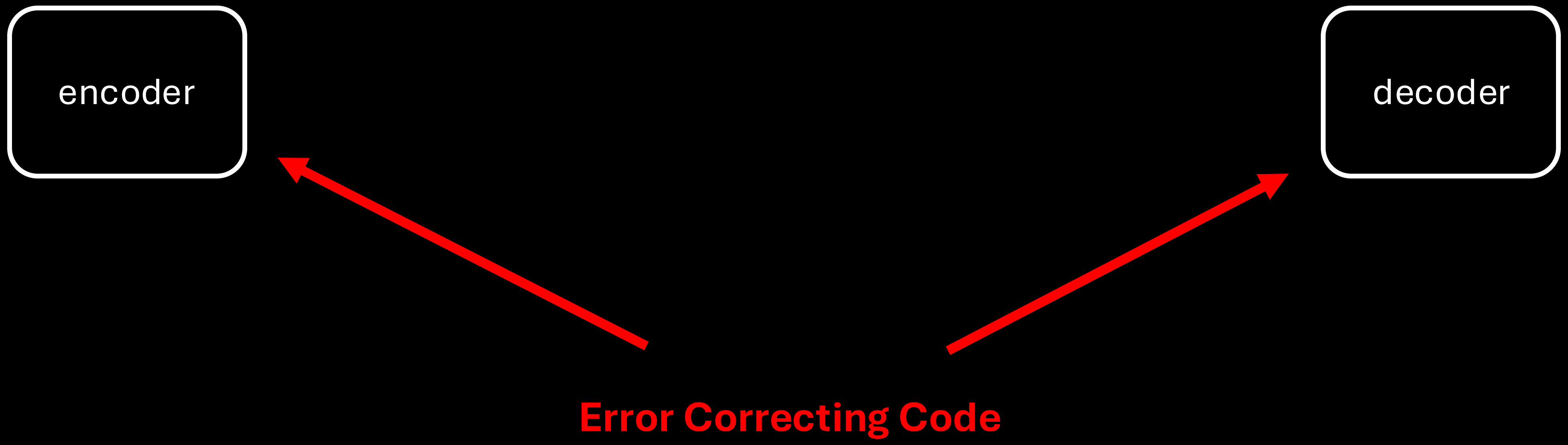
Background on Coding Theory



Background on Coding Theory



Background on Coding Theory



Linear Codes

Linear Codes

m

$\mathbb{F}_q^k$

message vector

Linear Codes

G

m

$\mathbb{F}_q^{k}$

$\mathbb{F}_q^{k \times n}$

message vector

generator matrix

Linear Codes

$$m \quad G \quad = \quad c$$

$\mathbb{F}_q^{\,k}$	$\mathbb{F}_q^{\,k \times n}$	$\mathbb{F}_q^{\,n}$
message vector	**generator matrix**	**codeword**

Linear Codes

the possible codewords form a subspace of $\mathbb{F}_q^n$

c_0

c_1

c_2

c_3

c_4

c_5

c_6

c_7

c_8

Linear Codes

2 important parameters

$R = k/n$

information rate

$D = d/n$

minimum distance

Linear Codes

2 important parameters

R = k/n

information rate

D = d/n

minimum distance

Linear Codes

2 important parameters

$R = k/n$

information rate

$D = d/n$

minimum distance

Singleton bound

$d \leq n - k + 1$

Reed-Solomon Codes

Reed-Solomon Codes

m

$m(x)$

$\mathbb{F}_q[x]$

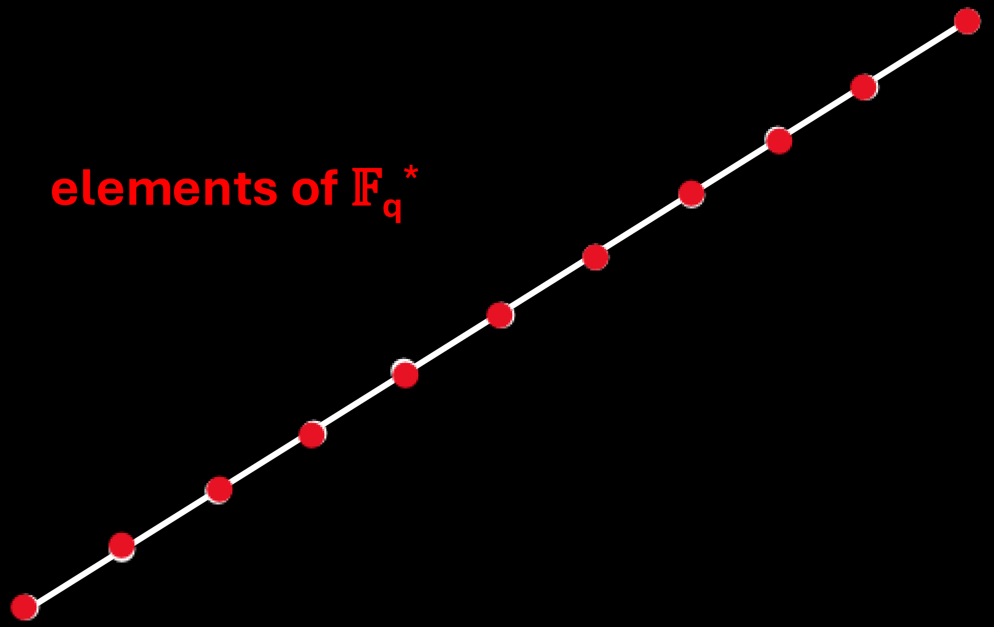
Reed-Solomon Codes

m

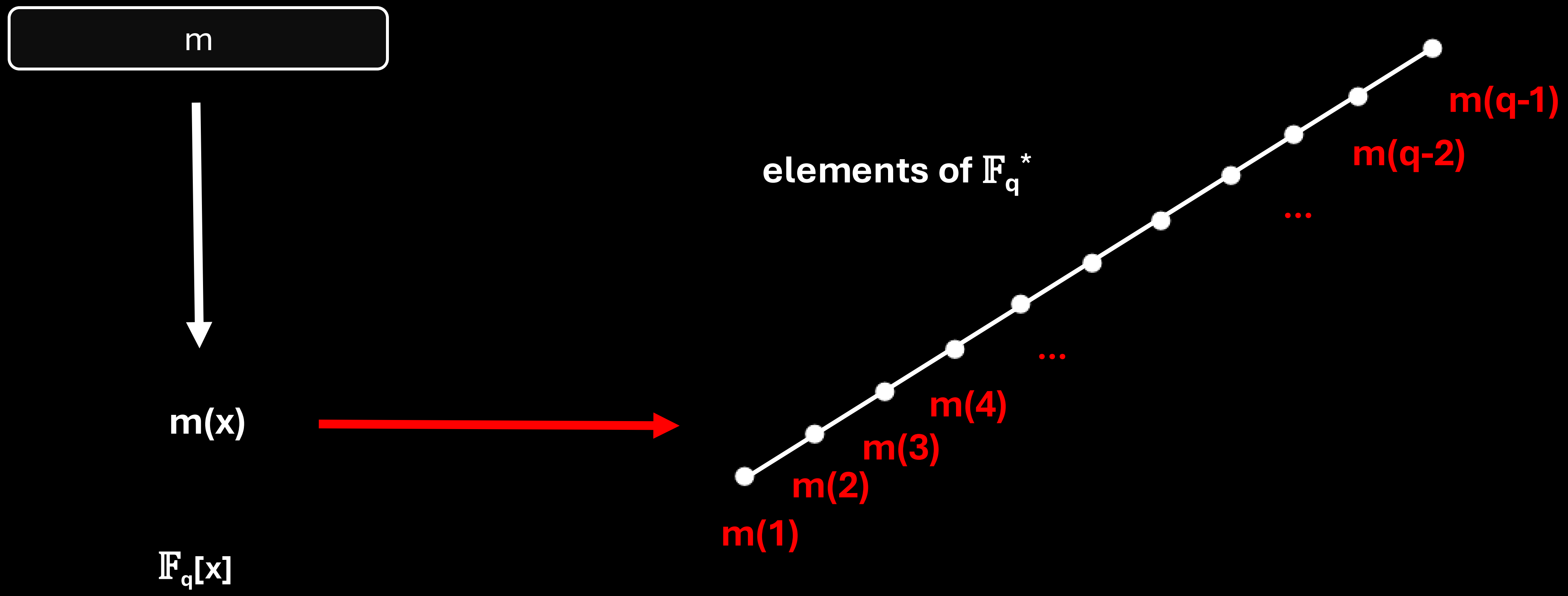
$\downarrow$

m(x)

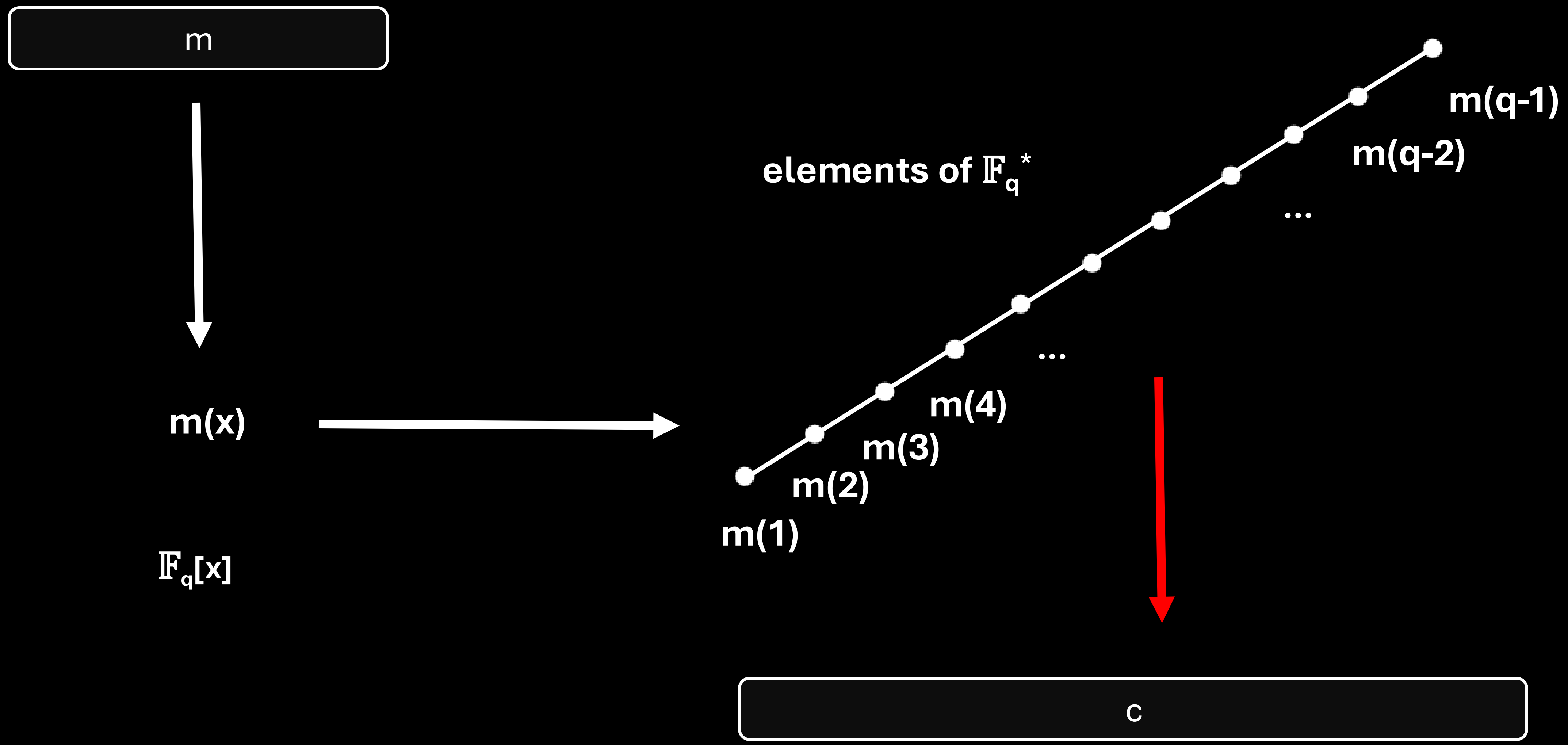
$\mathbb{F}_q[x]$



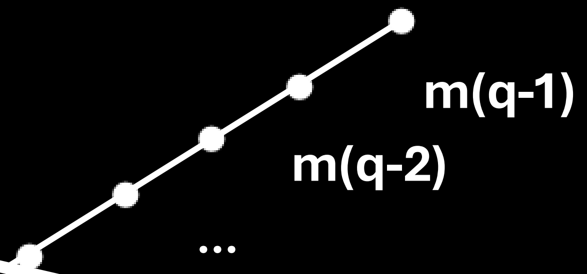
Reed-Solomon Codes



Reed-Solomon Codes



Reed-Solomon Codes



encoding = polynomial evaluation

$\mathbb{F}_q[x]$

c

Reed-Solomon Codes

pros:

MDS code $d = n - k + 1$

cons:

Reed-Solomon Codes

pros:

MDS code $d = n - k + 1$

linear code, easy en/decode

$$\begin{matrix} 1 & g & g^2 & g^3 & g^4 & \cdots \\ 1 & g^2 & g^4 & g^6 & g^8 & \cdots \\ 1 & g^3 & g^6 & g^9 & g^{12} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix}$$

Vandermonde matrix

cons:

Reed-Solomon Codes

pros:

MDS code $d = n - k + 1$

linear code, easy en/decode

$$\begin{matrix} 1 & g & g^2 & g^3 & g^4 & \cdots \\ 1 & g^2 & g^4 & g^6 & g^8 & \cdots \\ 1 & g^3 & g^6 & g^9 & g^{12} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix}$$

Vandermonde matrix

cons:

n (#evaluation points) < q (#field elements)

Reed-Solomon Codes

pros:

MDS code $\quad d = n - k + 1$

linear code, easy en/decode

$$\begin{matrix} 1 & g & g^2 & g^3 & g^4 & \cdots \\ 1 & g^2 & g^4 & g^6 & g^8 & \cdots \\ 1 & g^3 & g^6 & g^9 & g^{12} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix}$$

Vandermonde matrix

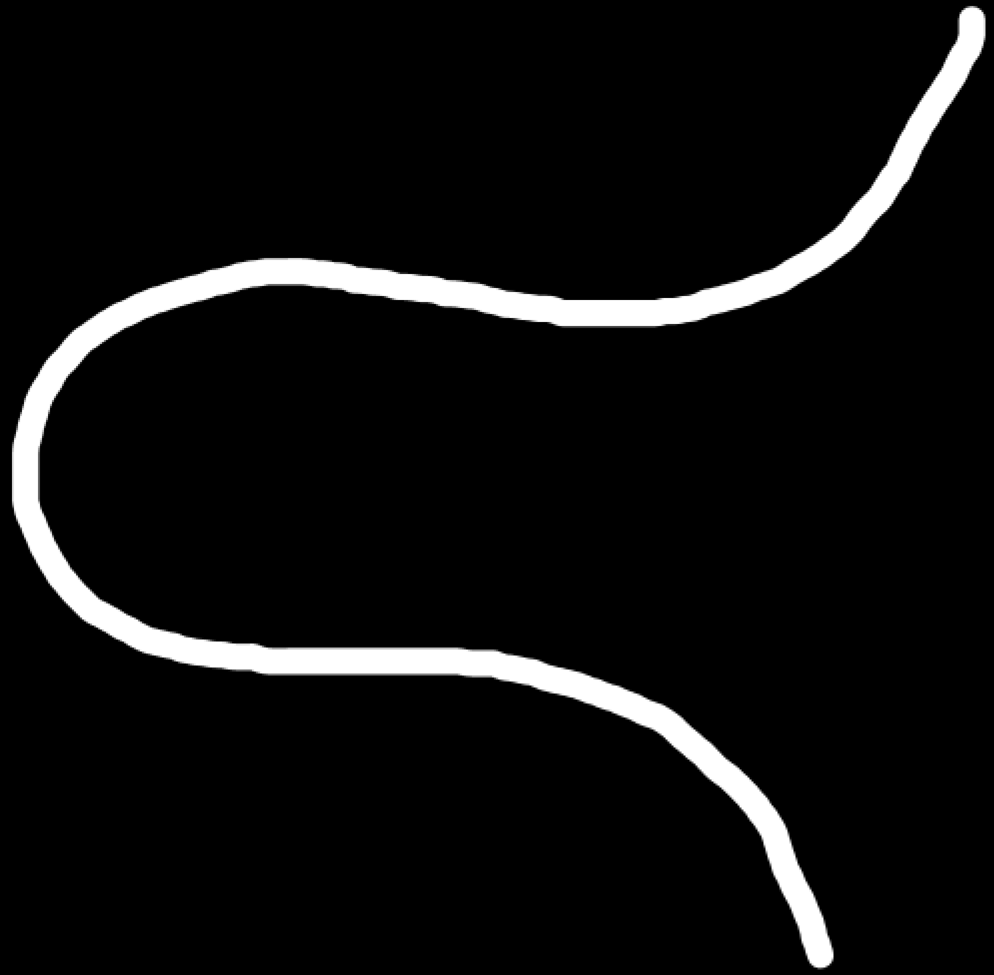
cons:

n (#evaluation points) < q (#field elements)

no asymptotically good family of codes as n → ∞

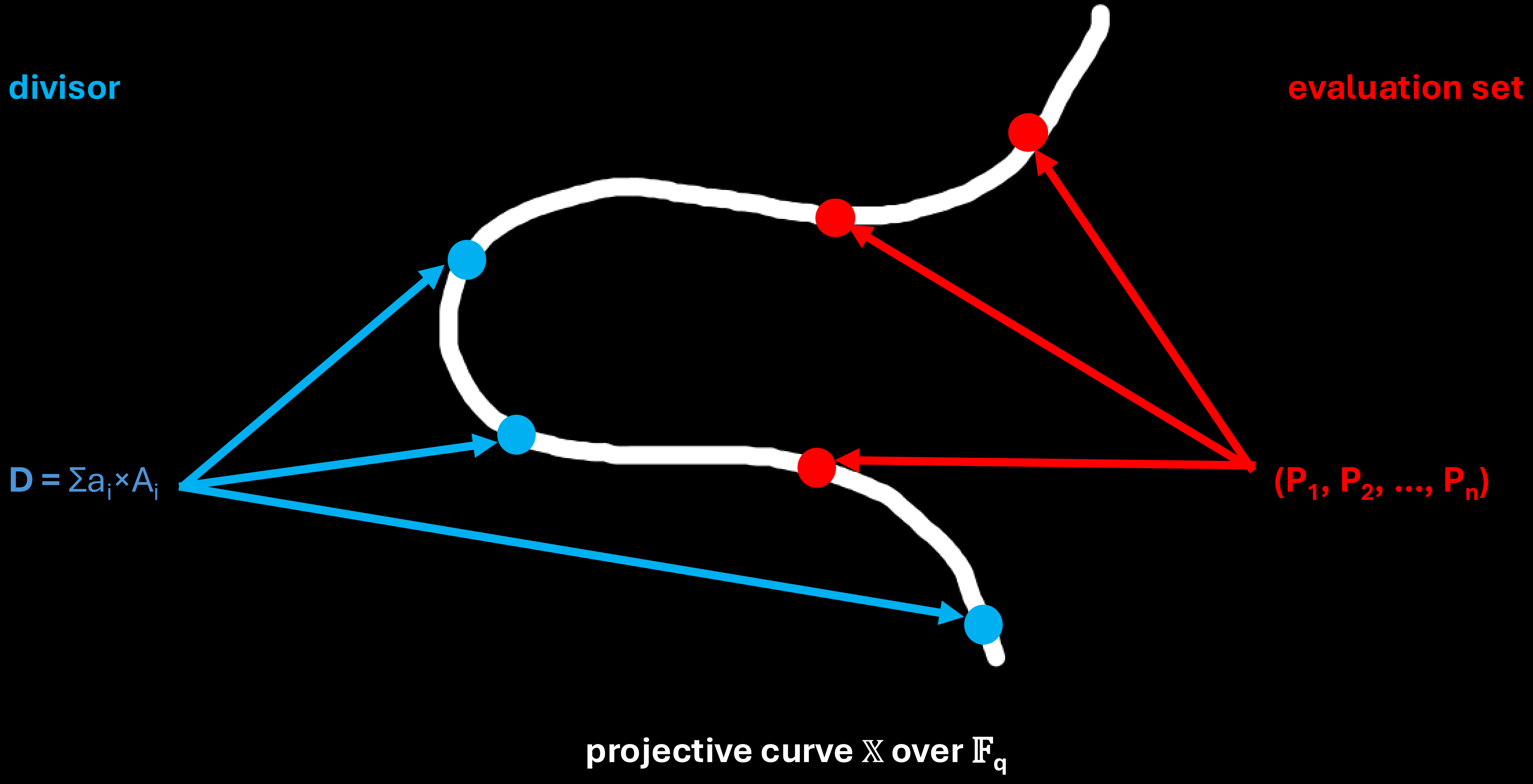
Algebraic Geometry Codes

Algebraic Geometry Codes



projective curve $\mathbb{X}$ over $\mathbb{F}_q$

Algebraic Geometry Codes



Algebraic Geometry Codes

Reed-Solomon **AG**

$$\mathbb{F}_q[x] \xrightarrow{\textbf{ambient function field}} F(\mathbb{X}) \sim \mathrm{Frac}(\mathbb{F}_q[x,y]/(\mathbb{X}))$$

Algebraic Geometry Codes

Reed-Solomon | **AG**

$\mathbb{F}_q[x] \xrightarrow{\text{ambient function field}} F(\mathbb{X}) \sim \mathrm{Frac}(\mathbb{F}_q[x,y]/(\mathbb{X}))$

$\mathbb{F}_q[x]_{<k} \xrightarrow{\text{space of messages}} \mathcal{L}(D)$ Riemann-Roch space of D

Algebraic Geometry Codes

Reed-Solomon **AG**

$$\mathbb{F}_q[x] \xrightarrow{\text{ambient function field}} F(\mathbb{X}) \sim \mathrm{Frac}(\mathbb{F}_q[x,y]/(\mathbb{X}))$$

$$\mathbb{F}_q[x]_{<k} \xrightarrow{\text{space of messages}} \mathcal{L}(D)$$

Riemann-Roch space of D

$$\text{elements of } \mathbb{F}_q \xrightarrow{\text{evaluation points}} (P_1, P_2, \ldots, P_n)$$

Algebraic Geometry Codes

Reed-Solomon		AG
$\mathbb{F}_q[x]$	ambient function field $\longrightarrow$	$F(\mathbb{X}) \sim \mathrm{Frac}(\mathbb{F}_q[x,y]/(\mathbb{X}))$
$\mathbb{F}_q[x]_{<k}$	space of messages $\longrightarrow$	$\mathcal{L}(D)$ Riemann-Roch space of D
elements of $\mathbb{F}_q$	evaluation points $\longrightarrow$	$(P_1, P_2, \ldots, P_n)$
MDS property	distance/dimension guarantee $\longrightarrow$	Riemann-(Roch) theorem
		$k = \dim(D) = \deg(D) + 1 - g(\mathbb{X})$
$d = n - k + 1$		$d = n - \deg(D)$

Code-based Public-Key Encryption

Code-based Public-Key Encryption

hard problem: decoding a linear code, where G is random

Code-based Public-Key Encryption

hard problem: decoding a linear code, where G is random

idea: create a G' that looks random, but we can decode

Code-based Public-Key Encryption

keygen:

1. generate randomly a structured G that we can decode
2. publish the masked G' := SGP
3. hope G' looks random to others

Code-based Public-Key Encryption

keygen:

1. generate randomly a structured G that we can decode
2. publish the masked $G' := SGP$
3. hope G' looks random to others

encryption:

1. encode the message and add noise: $c = mG' + e$

Code-based Public-Key Encryption

keygen:

1. generate randomly a structured G that we can decode
2. publish the masked G' := SGP
3. hope G' looks random to others

encryption:

1. encode the message and add noise: $c = mG' + e$

decryption:

1. permute back: $c' = c*P^{-1} = mSG + eP^{-1}$
2. decode: $dec(c') = mS$
3. multiply by S inverse: $m = mS*S^{-1}$

Code-based Public-Key Encryption

keygen:

1. generate randomly a structured G that we can decode
2. publish the masked G' := SGP
3. hope G' looks random to others

encryption:

1. encode the message and add noise: $c = mG' + e$

decryption:

1. permute back: $c' = c*P^{-1} = mSG + eP^{-1}$
2. decode: $\mathrm{dec}(c') = mS$
3. multiply by S inverse: $m = mS*S^{-1}$

only known secure instantiation uses AG codes, binary Goppa codes specifically