

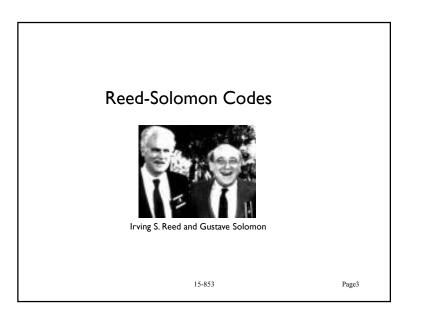
Decoding RS codes in polynomial time

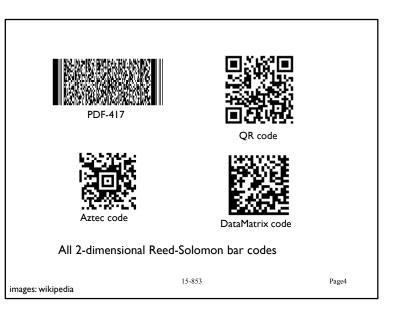
REED-SOLOMON DECODING

15-853

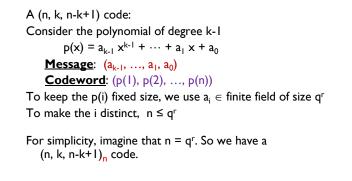
Page2

1









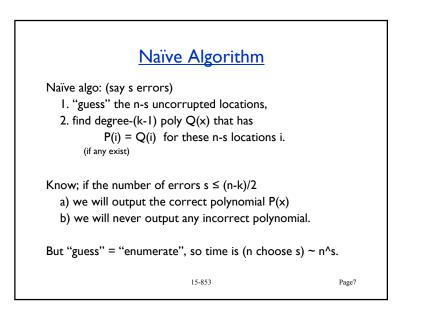
15-853

Page5

Encoding/Decoding Time

Can choose any n "interpolation points" E.g., choose n roots of unity Can then use FFT for encoding, take O(n log n) time. If there are no errors, can use FFT to decode the codeword, also O(n log n). If s errors, not clear what to do.

Page6



The Berlekamp Welch Algorithm

15-853

Say we sent $c_i = P(i)$ for i = 1..nReceived c'_i where $c_i = c'_i$ for all but s locations. Let S be the set of these s error locations.

Suppose we magically know error polynomial E(x)such that E(x) = 0 for all x in S. And E(x) has degree s.

Does such a thing exist?

Sure. E(x)

 $E(x) = \prod_{a \text{ in } S} (x - a)$

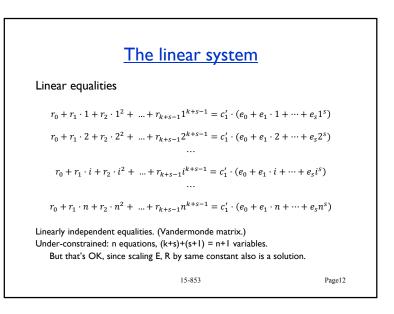
Page8

Say we sent $c_i = P(i)$ fo	or $i = 1n$	
Received c'_i where c_i	$= c'_i$ for all bu	t s locations.
Let S be the set of the	ese s error loc	ations.
Suppose we magically kn	now error poly	nomial $E(x)$
such that $E(x) = 0$ for	or all x in S.	
	or all x in S.	
And $E(x)$ has degree s.	or all x in S.	
such that $E(x) = 0$ for And $E(x)$ has degree s. Then we know that $P(i) \cdot E(i) = c'_i \cdot c'_i$		for all <i>i in</i> 1 <i>n</i>
And $E(x)$ has degree s. Then we know that		for all <i>i in</i> 1 <i>n</i>

The Berlekamp Welch Algorithm

Know that		
$P(i) \cdot E(i) = c'_i \cdot E(i)$	(i) for all <i>i</i> in	1 <i>n</i>
Want to solve for polys $P(x)$	x) (of deg $k - 1$), $E(x)$ of c	leg s.
How? First, rewrite as:		
$\mathbf{R}(i) = c'_i \cdot E(i)$	for all <i>i in</i> 1 <i>n</i>	
for polynomials R of degree	e (k+s-1), E of degree s.	
R has k+s "degrees of freed	lom". E has s+1.	
Have n equalities.		
So perhaps can get soluti	ion if $(k + s) + (s + 1) \ge$	n.
Return $\frac{R(x)}{E(x)}$.	15-853	Page10

<u>The current si</u>	ituation	
We know that $R(i) = c'_i \cdot E(i)$	for all <i>i</i> in 1n	
Suppose $R(x) = \sum_{j=1k+s-1} r_j x^j$ $k + s$ unknowns (the r_i values)		
And $E(x) = \sum_{j=0s} e_j x^j$ $s+1$ unknowns (the e_i values)		
How to solve for $R(x), E(x)$?		
15-853		Page11

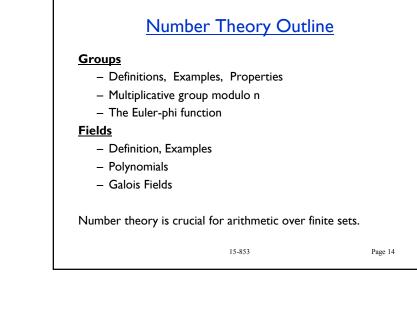


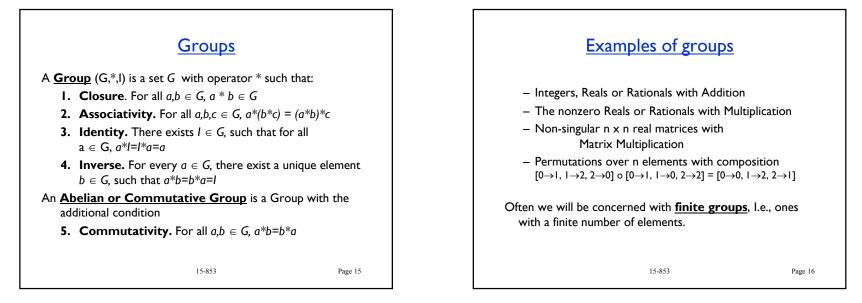
Math for both coding theory and cryptography

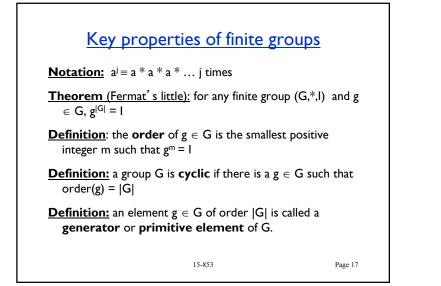
A NUMBER THEORY PRIMER

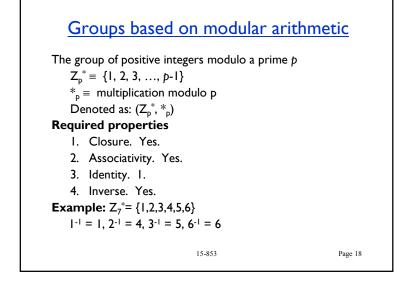
15-853

Page13

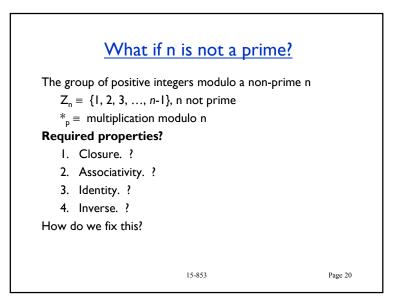


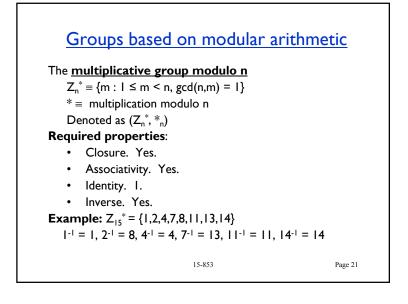


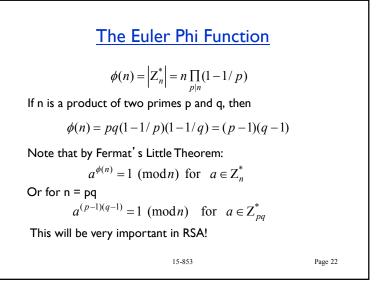


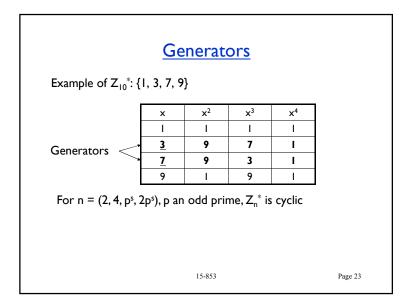


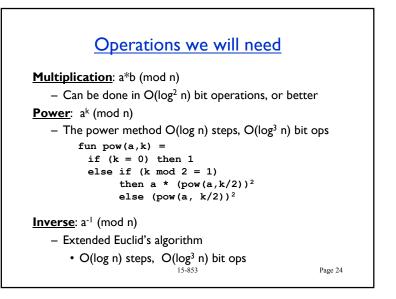
	<u>C</u>	<u>)the</u>	<u>r pro</u>	oper	<u>ties</u>		
Z _p * = (p-1) By Fermat's li Example of Z ₇		eoren	n: a ^{(p-1}) = 1 (mod p)	
	х	x ²	x ³	x ⁴	x ⁵	x ⁶	
	Ι	I	1	I	-	Ι	
	2	4	Ι	2	4	Ι	
,	3	2	6	4	5	I	
Generators <	4	2	I	4	2	Ι	
	5	4	6	2	3	Ι	
	6	-	6	Ι	6	Ι	
For all p the g	roup i	is cycli	C. 15-853	3			Page 19

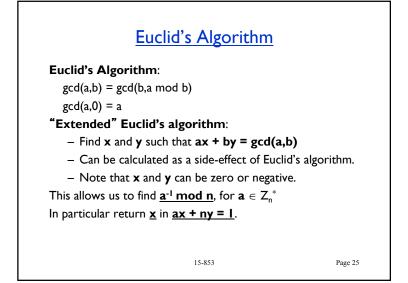


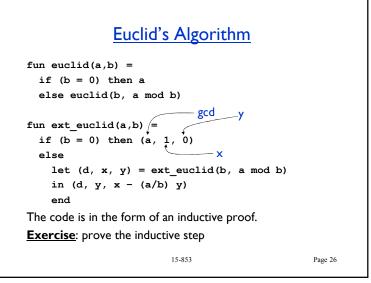


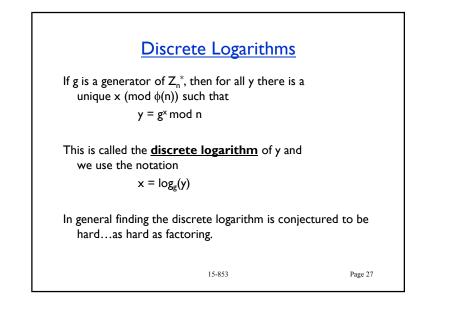


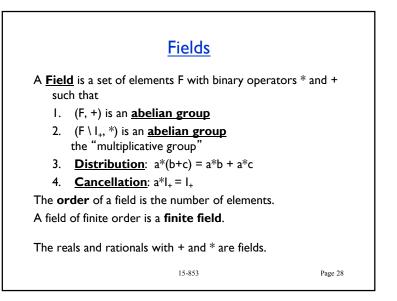


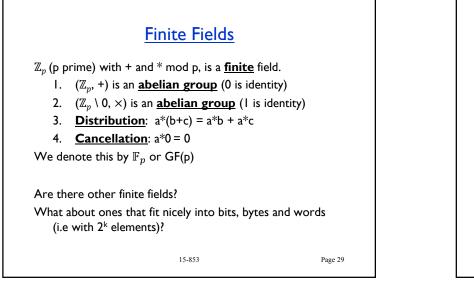


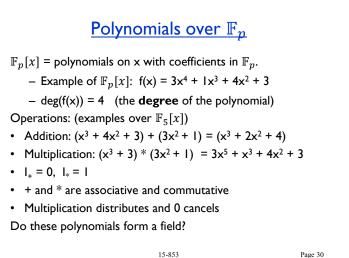


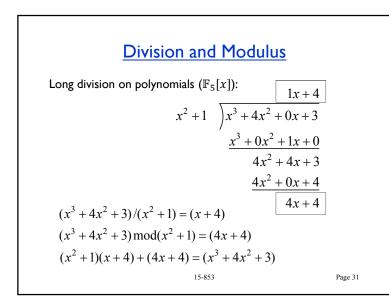


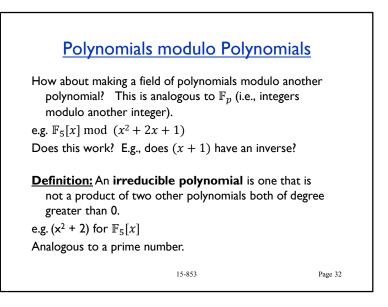












Galois Fields

The polynomials $\mathbb{F}_p[x] \mod p(x)$ where $p(x) \in \mathbb{F}_p[x]$, p(x) is irreducible, and $\deg(p(x)) = n$ (i.e. n+1 coefficients) form a finite field. Such a field has p^n elements.

These fields are called **Galois Fields** or **GF(p**ⁿ) or \mathbb{F}_{p^n} The special case n = 1 reduces to the fields \mathbb{F}_p . The special case p = 2 is especially useful for us.

15-853

Page 33

GF(2ⁿ)

 \mathbb{F}_{2^n} = set of polynomials in $\mathbb{F}_2[x]$ modulo irreducible polynomial $p(x) \in \mathbb{F}_2[x]$ of degree n.

Elements are all polynomials in $\mathbb{F}_2[x]$ of degree $\leq n-1$. Has 2^n elements. Natural correspondence with bits in $\{0,1\}^n$.

E.g., $x^6 + x^4 + x + 1 = 01010011$

Elements of \mathbb{F}_{2^8} can be represented as ${\bf a}$ byte, one bit for each term.

15-853

Page 34

