MODULAR ARITHMETIC

- DIVIDING AN INTEGER a BY ANOTHER INTEGER b CAN LEAVE A „MODULUS", OR „REMAINDER" m; WE WRITE

$$a \bmod b = m$$

OR

(1) $$a \equiv m \pmod b$$

- EXAMPLES:
  - $15 \bmod 7 = 1$ SINCE $15 = 2 \cdot 7 + 1$
  - $3275 \bmod 256 = 203$ SINCE $3275 = 12 \cdot 256 + 203$
  - $(-9) \bmod 7 = 5$ SINCE $-9 = -2 \cdot 7 + 5$
  - $21 \bmod 7 = 0$ SINCE $21 = 3 \cdot 7$

- THE MODULUS CAN ALWAYS BE ASSUMED TO BE A POSITIVE INTEGER STRICTLY SMALLER THAN b, E.G. WE COULD WRITE $15 \bmod 7 = 8$ SINCE $15 = 1 \cdot 7 + 8$, BUT THEN WE GET $15 = 1 \cdot 7 + 8 = 1 \cdot 7 + 7 + 1 = 2 \cdot 7 + 1$

- IF TWO NUMBERS a AND c LEAVE THE SAME MODULUS WHEN DIVIDING BY b, WE SAY THAT a AND c ARE „CONGRUENT", OR „EQUIVALENT", OR „EQUAL MODULO b"

- E.G. 15 AND 8 ARE EQUAL MODULO 7, WE WRITE

$$15 \equiv 8 \pmod 7$$

- THIS IS THE SAME NOTATION AS (1): ANY INTEGER a IS CLEARLY EQUAL TO ITS 'REMAINDER MODULO ~~b~~' MODULO b

- MODULAR ARITHMETIC AMOUNTS TO APPLYING THE USUAL ARITHMETICAL OPERATIONS E.G. ADDITION, SUBTRACTION, EXPONENTIATION BUT COMPUTING THE MODULUS MODULO b AFTER EVERY OPERATION

- WHEN WORKING MODULO b WE THUS ONLY NEED TO CONSIDER THE POSSIBLE REMAINDERS MODULO b, AND NOT „ALL" INTEGERS

- THE POSSIBLE REMAINDERS MODULO b ARE DENOTED BY $\mathbb{Z}_b$, E.G. $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$, $\mathbb{Z}_2 = \{0, 1\}$

- EXAMPLES:
  - $7 \cdot 8 \equiv 8 \bmod 12$
  - $1 - 2 \equiv 6 \bmod 7$
  - $2^8 \equiv 1 \bmod 5$
  - $14 + 1 \equiv 0 \bmod 15$

· ALTERNATIVE VIEW: THE NUMBERS „WRAP AROUND" IF THEY ARE TOO HIGH ($>b$) OR LOW (<0)

· THE ORDER IN WHICH WE COMPUTE AND MODULIZE DOES NOT MATTER, E.G.

$$58 \cdot 13 = 754 \equiv 5 \bmod 7$$
$$58 \equiv 2 \bmod 7$$
$$13 \equiv 6 \bmod 7$$
$$2 \cdot 6 = 12 \equiv 5 \bmod 7$$

· THIS IS USEFUL IN CRYPTOGRAPHY BECAUSE A MESSAGE CAN BE WRITTEN AS A SEQUENCE OF NUMBERS. THEN ENCRYPTION AND DECRYPTION CAN BE IMPLEMENTED BY APPLYING OPERATIONS TO THESE NUMBERS.

· EXAMPLE: THE CAESAR/SHIFT CIPHER
  - THE MESSAGE IS MADE UP OF LETTERS
  - THERE ARE 26 POSSIBLE LETTERS
  - IDENTIFY THEM WITH THE INTEGERS IN $\mathbb{Z}_{26}$ E.G. $A=0$, $B=1$, $C=2$ ... $Z=25$. „HELLO WORLD" THEN BECOMES „7-4-11-11-14" 22-14-17-11-3
  - SUPPOSE THE KEY IS $k=10$. THEN ENCRYPTION CAN BE WRITTEN AS

$$E(x, k) = (x+k) \bmod 26$$

AND DECRYPTION AS

$$D(x, k) = (x-k) \bmod 26$$

SO THAT E.G. $E(7, 10) = 17$, $E(4, 10) = 14$, $E(22, 10) = 6$, ETC., GIVING THE CIPHERTEXT 17-14-21-21-24-6-24-1-21-13, I.E. „ROVVY GYBVN"

AFFINE CIPHER

· AGAIN, CONSIDER $\mathbb{Z}_{26}$

· THE KEY IS A PAIR (a, b) OF NUMBERS FROM $\mathbb{Z}_{26}$

· ENCRYPTION IS DEFINED BY

$$E(x, (a,b)) = a \cdot x + b \bmod 26$$

· THEN DECRYPTION „SHOULD BE"

$$D(y, (a,b)) = (y-b)a^{-1} \bmod 26$$

SINCE $((a \cdot x + b) - b)a^{-1} = (a \cdot x)a^{-1} = a \cdot a^{-1} \cdot x = x$

- BUT DIVISION IS NOT ALWAYS POSSIBLE MODULO 26!
- "DIVIDING" a BY b SIMPLY MEANS MULTIPLYING a BY b^{-1} WHICH IS CALLED THE "INVERSE" OF b. IN OTHER WORDS, THE "INVERSE" OF $b \in \mathbb{Z}_n$ IS SOME ELEMENT $b^{-1} \in \mathbb{Z}_n$ SUCH THAT $b \cdot b^{-1} \equiv 1 \bmod n$
- FOR EXAMPLE, $3^{-1} \equiv 9 \bmod 26$ SINCE $3 \cdot 9 = 27 \equiv 1 \bmod 26$. BUT 0 HAS NO INVERSE, AND NEITHER DOES E.G. 4: NO MATTER WHAT WE MULTIPLY IT BY, WE NEVER GET 1; IN FACT, ONLY 12 ELEMENTS OF $\mathbb{Z}_{26}$ ARE INVERTIBLE, VIZ. 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25
- SO a IN THE AFFINE CIPHER MUST BE CHOSEN FROM AMONG THESE 12 ELEMENTS. THERE IS NO RESTRICTION ON b, SO THERE ARE $12 \cdot 26 = 312$ POSSIBLE KEYS
- EXAMPLE: $k = (a, b) = (7, 11)$. THEN E.G. $E(5, k) = 5 \cdot 7 + 11 \bmod 26 = 20$, SO "F" ENCRYPTS TO "U". DECRYPTING $D(20, k) = (20 - 11) 7^{-1} = 9 \cdot 7^{-1} = 9 \cdot 15 = 5 \bmod 26$ SO "U" DECRYPTS BACK TO "F"
- INSTEAD OF WORKING WITH NUMBERS MODULO 26 TO ENCRYPT LETTERS WE MAY WORK MODULO 256 AND ENCRYPT SEQUENCES OF BYTES, MODULO 36 TO INCLUDE DIGITS, ETC.
- ANY NON-TRIVIAL CIPHER WILL INVOLVE MULTIPLICATION AND, HENCE, DIVISION. WE THUS NEED TO KNOW WHICH ELEMENTS OF $\mathbb{Z}_n$ ARE INVERTIBLE FOR ANY GIVEN n.

MORE NUMBER THEORY

- WE SAY THAT a "DIVIDES" b, WRITE $a | b$, IF $b \equiv 0 \bmod a$, I.E. "b IS A MULTIPLE OF a"
- EXAMPLE: SHOW THAT $a \equiv b \bmod n$ IFF $n | a - b$
- IF $a | b$ AND $c | b$, WE SAY THAT b IS A "COMMON MULTIPLE" OF a AND b
- IF $a | b$ AND $a | c$ WE SAY THAT a IS A "COMMON DIVISOR" OF b AND c
- BY $\gcd(a, b)$, RESP. $\operatorname{lcm}(a, b)$ WE DENOTE THE GREATEST COMMON DIVISOR, RESP. LEAST COMMON MULTIPLE OF a AND b
- EXAMPLE: $\gcd(15, 9) = 3$, $\gcd(321, 27) = 3$, $\operatorname{lcm}(15, 9) = 45$, $\operatorname{lcm}(321, 27) = 2889$

THEOREM | $a \in \mathbb{Z}_n$ IS INVERTIBLE IFF $\gcd(a, n) = 1$

- THE GCD OF TWO INTEGERS CAN BE COMPUTED USING THE „EUCLIDEAN ALGORITHM"

EUCLIDEAN ALGORITHM

INPUT: $a, b \in \mathbb{Z}$, $a > b$
1) WRITE $a = q \cdot b + r$ FOR $r < b$
2) IF $r = 0$ RETURN b
3) SET $a \leftarrow b$, $b \leftarrow r$ AND GO TO STEP 1)

- EXAMPLE: $GCD(17171, 2442) = 11$:

$$\begin{aligned} 17171 &= 7 \cdot 2442 + 77 \\ 2442 &= 31 \cdot 77 + 55 \\ 77 &= 1 \cdot 55 + 22 \\ 55 &= 2 \cdot 22 + 11 \\ 22 &= 2 \cdot 11 \end{aligned}$$

- GOING BACKWARDS" FROM BOTTOM TO TOP, WE CAN EXPRESS 11 AS

$$\begin{aligned} 11 &= 55 - 2 \cdot 22 = 55 - 2(77 - 55) = -2 \cdot 77 + 3 \cdot 55 = \\ &= -2 \cdot 77 + 3(2442 - 31 \cdot 77) = 3 \cdot 2442 - 95 \cdot 77 = \\ &= 3 \cdot 2442 - 95(17171 - 7 \cdot 2442) = -95 \cdot 17171 + \\ & 668 \cdot 2442 \end{aligned}$$

- THIS IS A SPECIAL CASE OF A MORE GENERAL PHENOMENON

THEOREM | LET $a, b \in \mathbb{Z}$ WITH $ab \neq 0$. THEN THERE EXIST $\alpha, \beta \in \mathbb{Z}$ SUCH THAT

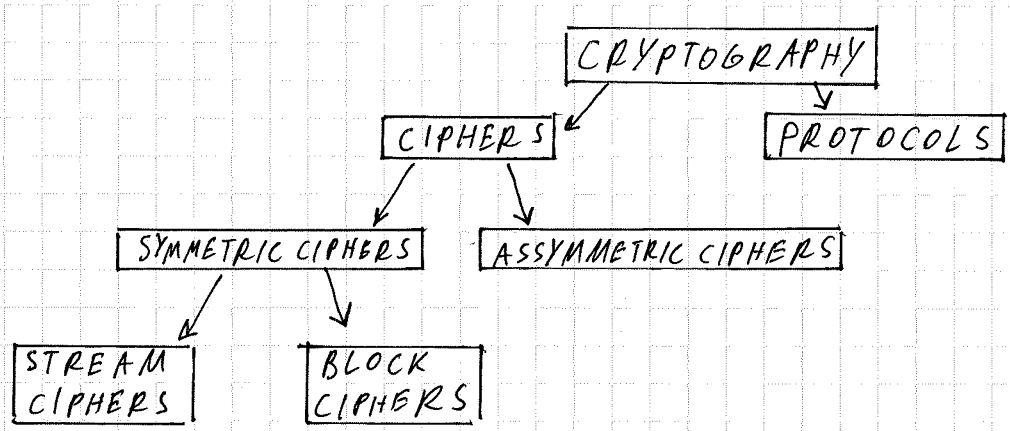
$$\alpha a + \beta b = \gcd(a, b).$$

- α AND β ABOVE ARE CALLED „BEZOUT COEFFICIENTS"

- USING THIS WE CAN PROVE THE FIRST THEOREM:
  "$\Rightarrow$": IF $a \in \mathbb{Z}_n$ IS INVERTIBLE, THEN $\exists \alpha \in \mathbb{Z}_n$ S.T. $a \cdot \alpha \equiv 1 \bmod n$ I.E. $a \cdot \alpha = n \cdot \beta + 1$ FOR SOME $\beta \in \mathbb{Z}$. SUPPOSE d IS A COMMON DIVISOR OF a AND n. THEN $a = \gamma_1 d$ AND $n = \gamma_2 d$ FOR SOME $\gamma_1, \gamma_2 \in \mathbb{Z}$. HENCE $\alpha \gamma_1 d = \beta \gamma_2 d + 1$, I.E. $(\alpha \gamma_1 - \beta \gamma_2) d = 1$, I.E. $d \mid 1$, I.E. $d = 1$.
  "$\Leftarrow$": IF $\gcd(a, n) = 1$, THEN THERE $\alpha, \beta \in \mathbb{Z}$ S.T. $\alpha a + \beta n = 1$, I.E. $\alpha a = \beta n + 1 \equiv 1 \bmod n$. THUS $\alpha \bmod n$ IS THE INVERSE OF a. $\square$

- MODULAR ARITHMETIC IS VERY IMPORTANT FOR THE STUDY OF I.A. CRYPTOGRAPHY

A MAP OF CRYPTOGRAPHY



- DATA REPRESENTATION: MESSAGES CAN BE WRITTEN IN LETTERS, NUMBERS, BITS, ETC. UNTIL FURTHER NOTICE WE WILL ASSUME THAT THESE „SYMBOLS" ARE BITS

STREAM CIPHERS

- STREAM CIPHERS ENCRYPT THE PLAINTEXT SYMBOL BY SYMBOL (BIT BY BIT)
- THIS ACHIEVED BY GENERATING A „KEYSTREAM" FROM THE SECRET KEY. THE KEYSTREAM CAN BE ARBITRARILY LONG AND IT DEPENDS DETERMINISTICALLY ON THE KEY, I.E. THE SAME KEY WILL ALWAYS GENERATE THE SAME KEYSTREAM. THE BITS OF THE PLAINTEXT ARE THEN COMBINED WITH THOSE OF THE KEYSTREAM (LIKE IN THE VERNAM CIPHER) TO PRODUCE THE CIPHERTEXT.
- (VERY BAD) EXAMPLE: THE KEYSTREAM IS GENERATED BY REPEATING THE KEY OVER AND OVER AGAIN FOR AS LONG AS NEEDED

$$K = 0100 \qquad \begin{array}{rl} M = & 0\,1\,1\,0\,1\,1\,1\,0\,0\,1\,1\,1\,0\,0\,0 \\ \text{KEYSTREAM} = & 0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0 \\ \hline C = & 0\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,1\,0\,1\,0 \end{array}$$

- STREAM CIPHERS CAN BE „SYNCHRONOUS" AND „ASYNCHRONOUS". IN THE FORMER, THE KEYSTREAM ONLY DEPENDS ON THE KEY. IN THE LATTER, THE KEYSTREAM CAN ALSO DEPEND ON THE PREVIOUSLY PROCESSED CIPHERTEXT.
- BLOCK CIPHERS ENCRYPT THE PLAINTEXT ONE „BLOCK" E.G. 128 BITS AT A TIME. WHEN THE BLOCK IS BEING ENCRYPTED, EVERY BIT OF THE OUTPUT BLOCK CAN POTENTIALLY DEPEND ON EVERY BIT OF THE INPUT BLOCK, WHICH MAKES IT EASY TO DESIGN COMPLICATED CIPHERS.

- STREAM CIPHERS ARE TYPICALLY VERY LIGHTWEIGHT, AND ARE USED IN ENVIRONMENTS WITH LIMITED COMPUTATIONAL RESOURCES, E.G. CREDIT CARDS, CELL PHONES
- BLOCK CIPHERS ARE MORE POPULAR FOR ENCRYPTING COMPUTER COMMUNICATIONS
- MODERN BLOCK CIPHERS CAN BE AS EFFICIENT AS STREAM CIPHERS
- ENCRYPTION AND DECRYPTION WITH STREAM CIPHERS IS ALWAYS THE SAME: IF THE PLAINTEXT IS $x_1x_2 \cdots x_n$, THE CIPHERTEXT IS $y_1y_2 \cdots y_n$, AND THE KEYSTREAM IS $k_1k_2 \cdots k_n$, THEN
$$\forall i \quad y_i = x_i + k_i \mod 2$$
$$\forall i \quad x_i = y_i + k_i \mod 2$$
- THE QUESTION IS HOW TO GENERATE THE KEYSTREAM FROM THE KEY
- NOTE THAT THE VERNAM CIPHER IS A STREAM CIPHER; WE ARE LOOKING FOR MORE EFFICIENT WAYS OF GENERATING THE KEYSTREAM

RANDOM NUMBER GENERATORS

- IN ORDER FOR THE CIPHER TO BE SECURE THE KEYSTREAM SHOULD NOT HAVE ANY STATISTICAL PATTERNS OR DEPENDENCIES E.G. THE FREQUENCIES OF 0's AND 1's SHOULD BE THE SAME, ETC.
- "RANDOM NUMBER GENERATORS (RNG'S)" AIM TO PRODUCE SEQUENCES OF BITS, OR NUMBERS, WITH THOSE SAME PROPERTIES
- "TRUE RNG'S" (TRNG'S) GENERATE SEQUENCES OF NUMBERS OR BITS WHICH CANNOT BE REPRODUCED E.G. FLIPPING A COIN. TRNG'S CAN BE USED FOR GENERATING KEYS, BUT NOT FOR GENERATING THE KEYSTREAM.
- "PSEUDORANDOM NUMBER GENERATORS (PRNG'S)" GENERATE SEQUENCES THAT ARE COMPUTED DETERMINISTICALLY FROM AN INITIAL "SEED" VALUE E.G. THE rand() FUNCTION IN ANSI C GENERATES NUMBERS VIA
$$S_{i+1} = 1103515245\, S_i + 12345 \mod 2^{31}$$
WITH S_0 BEING THE SEED VALUE
- "CRYPTOGRAPHICALLY SECURE PRNG'S (CSPRNG'S)" ARE PRNG'S WHOSE OUTPUT IS "UNPREDICTABLE",

MEANING THAT IF ONE KNOWS $S_i, S_{i+1}, \ldots, S_{n+i-1}$ THERE IS NO POLYTIME ALGORITHM ABLE TO PREDICT THE NEXT BIT S_{n+i} WITH PROBABILITY BETTER THAN 1/2. IN ADDITION, IT SHOULD BE COMPUTATIONALLY INFEASIBLE TO FIND S_{i-1}, S_{i-2}, ETC.

- (BAD) EXAMPLE: USING A PRNG TO GENERATE A KEYSTREAM.
  - CONSIDER A PRNG GENERATING NUMBERS VIA

$$S_{i+1} \equiv AS_i + B \mod m$$

WITH S_0 BEING THE SEED, m CHOSEN TO HAVE E.G. 100 BITS, AND $A, B, S_i \in \mathbb{Z}_m$

  - THE KEY IS (A, B, S_0)
  - THE KEYSTREAM IS OBTAINED BY CONCATENATING THE BITS OF S_0, S_1, S_2, S_3 ETC. FOR EXAMPLE GENERATING THREE NUMBERS PRODUCES A KEYSTREAM OF 300 BITS
  - THE KEY CAN EASILY BE RECOVERED VIA A KNOWN PLAINTEXT ATTACK: IF THE CRYPTANALYST HAS 300 BITS OF PLAINTEXT AND CORRESPONDING CIPHERTEXT, HE CAN OBTAIN 300 BITS OF THE KEYSTREAM, I.E. HE KNOWS S_0, S_1, S_2
  - WE NOW WRITE:

$$S_2 \equiv AS_1 + B \mod m$$
$$S_3 \equiv AS_2 + B \mod m$$

HENCE

$$S_2 - S_3 \equiv A(S_1 - S_2) \mod m$$

SO THAT

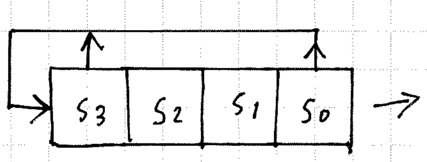
(2) $$A \equiv \frac{S_2 - S_3}{S_1 - S_2} \mod m$$

  - ONCE WE KNOW A, WE COMPUTE $B \equiv S_2 - AS_1 \mod m$
  - IF $\gcd((S_1 - S_2), m) \neq 1$, WE CAN GET SEVERAL VALUES OF A SATISFYING (2). KNOWLEDGE OF ADDITIONAL PLAINTEXT-CIPHERTEXT PAIRS CAN BE USED TO RULE OUT FALSE POSITIVES

- THE RNG ABOVE WAS A "GOOD" PRNG BUT NOT A CSPRNG. IF THE GENERATED SEQUENCE OF NUMBERS IS UNPREDICTABLE, AN ATTACK LIKE ABOVE WILL NOT WORK.

LINEAR FEEDBACK SHIFT REGISTERS

- A LINEAR FEEDBACK SHIFT REGISTER (LFSR) IS A CONCEPTUAL "DEVICE" WHICH PRODUCES A LONG PSEUDORANDOM "SEQUENCE OF BITS. LFSR'S ARE USED IN PRACTICE IN A LOT OF STREAM CIPHERS.



- AN LFSR OF DEGREE m CONSISTS OF m "CELLS" $s_0, s_1, \ldots, s_{m-1}$ AND A "FEEDBACK LINE" TO WHICH SOME OF THE CELLS ARE CONNECTED

- ABOVE IS AN LFSR OF DEGREE $m = 4$ WITH ITS FIRST AND LAST CELLS, s_0 AND s_3, CONNECTED TO THE FEEDBACK LINE

- THE CELLS ARE INITIALIZED WITH m BITS, E.G. $(s_3, s_2, s_1, s_0) = (1, 1, 1, 0)$

- AT EVERY "STEP" THE LFSR ACTS AS FOLLOWS:
  - ALL "CELLS CONNECTED TO THE FEEDBACK LINE ARE SUMMED MODULO 2
  - THIS SUM IS RECORDED IN THE LAST CELL s_{m-1}
  - THE VALUE PREVIOUSLY AT s_{m-1} MOVES TO s_{m-2}
  - s_{m-2} GOES TO s_{m-3}
  - ...
  - s_1 GOES TO s_0
  - s_0 IS EMITTED AS OUTPUT

- EXAMPLE: 1110 → 1111 → 0111 → 1011 →
  0101 → 1010 → 1101 → 0110 →
  0011 → 1001 → 0100 → 0010 →
  0001 → 1000 → 1100 → 1110

  011110101100100 0

- AN LFSR MUST EVENTUALLY LOOP

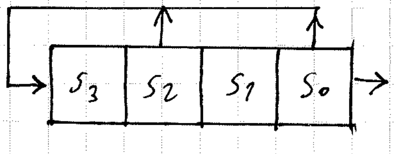
- THE VALUES OF ALL THE CELLS AT ANY GIVEN TIME IS THE "STATE" OF THE LFSR, E.G. THE INITIAL STATE IN THE "ABOVE EXAMPLE IS (1, 1, 1, 0)

- AN LFSR OF DEGREE m HAS 2^m DIFFERENT STATES

- AN LFSR OF DEGREE m CAN PRODUCE A SEQUENCE OF STATES/BITS OF LENGTH AT MOST $(2^m - 1)$ BEFORE IT LOOPS

- E.G. WITH 4 BITS WE CAN GENERATE 15 STATES, WITH 16 BITS WE CAN GENERATE 65 535 BITS!

- IS IT POSSIBLE THAT SOME LFSR WILL START LOOPING SOONER, I.E. WILL PRODUCE A SHORTER SEQUENCE?



1 1 1 0 → 1 1 1 1 → 0 1 1 1 →
0 0 1 1 → 1 0 0 1 → 1 1 0 0 →
1 1 1 0 [0 1 1 1 1 0]

- REPETITION AND LOOPING WITHIN THE KEYSTREAM ARE BAD FOR CRYPTOGRAPHIC APPLICATIONS

- HOW CAN WE DETERMINE WHEN AN LFSR WILL GIVE A „FULL-LENGTH" SEQUENCE?

- LFSR'S CAN BE REPRESENTED AS POLYNOMIALS: A DEGREE m LFSR CAN BE IDENTIFIED WITH THE POLYNOMIAL $p(x)$ GIVEN BY

(3) $$p(x) = x^m + \sum_{i=0}^{m-1} a_{m-i} x^{m-i}$$

WITH a_i BEING 1 IF S_i IS CONNECTED TO THE FEEDBACK LINE, AND a_i BEING 0 OTHERWISE

- THE ABOVE TWO LFSR'S CORRESPOND TO x^4+x^3+1 AND x^4+x^2+1

- THIS IS SIMPLY AN ALTERNATIVE WAY OF REPRESENTING THE LFSR

- IN ALGEBRA A POLYNOMIAL $p(x)$ IS CALLED „PRIMITIVE" IF IT IS THE MINIMAL POLYNOMIAL OF A PRIMITIVE ELEMENT OF THE EXTENSION FIELD $\mathbb{F}_{p^m}$

<u>THEOREM</u> | AN LFSR PRODUCES A SEQUENCE OF MAXIMAL POSSIBLE LENGTH IF AND ONLY IF ITS ASSOCIATED POLYNOMIAL IS PRIMITIVE.

- PRIMITIVE POLYNOMIALS CAN EASILY BE FOUND IN PRACTICE BY CONSULTING TABLES OR USING SPECIALIZED MATHEMATICAL SOFTWARE

ATTACKING AN LFSR

- A STREAM CIPHER USING THE OUTPUT SEQUENCE OF AN LFSR AS A KEYSTREAM IS VULNERABLE TO A KNOWN PLAINTEXT ATTACK

- THE ATTACKER KNOWS PART OF THE PLAINTEXT AND CORRESPONDING CIPHERTEXT AND CAN THEREFORE RECOVER PART OF THE KEYSTREAM. THE GOAL IS TO „RECONSTRUCT" THE LFSR, I.E. FIND ITS LENGTH m AND FIND WHICH

CELLS ARE CONNECTED TO THE FEEDBACK LINE

- IF THE OUTPUT OF THE LFSR (THE KEYSTREAM) IS DENOTED BY $S_0 S_1 \dots$ AND $a_0 a_1 \dots$ INDICATE WHETHER THE CELLS 'ARE CONNECTED' TO THE FEEDBACK AS IN (3), WE CAN TRY TO GUESS m AND WRITE

$$S_{n+m} \equiv a_0 S_n + a_1 S_{n+1} + a_2 S_{n+2} + \dots + a_{m-1} S_{n+m-1} \mod 2$$

OR

$$S_{n+m} \equiv \sum_{i=0}^{m-1} a_i S_{n+i}$$

- WE GUESS $m=2$ AND WRITE (ASSUMING THAT WE KNOW $S_n, S_{n+1}, S_{n+2}, S_{n+3}$)

(4)
$$\begin{aligned} S_{n+2} &\equiv a_0 S_n + a_1 S_{n+1} \\ S_{n+3} &\equiv a_0 S_{n+1} + a_1 S_{n+2} \end{aligned}$$

- WE KNOW S_i E.G. 01101011100 SO $S_n = 0 = S_{n+3}$, $S_{n+1} = 1 = S_{n+2}$. THE SYSTEM (4) BECOMES

$$\begin{aligned} 1 &\equiv 0 \cdot a_0 + 1 \cdot a_1 \\ 0 &\equiv 1 \cdot a_0 + 1 \cdot a_1 \end{aligned}$$

I.E.

$$\begin{aligned} 1 &\equiv a_1 \\ 0 &\equiv a_0 + a_1 \end{aligned}$$

HENCE $a_0 \equiv 1$, $a_1 \equiv 1$. THEN $S_{n+4} = 1 \cdot S_{n+2} + 1 \cdot S_{n+3} \equiv 1+0 = 1$, $S_{n+5} = 1 \cdot S_{n+3} + 1 \cdot S_{n+4} \equiv 0+1 \equiv 1$ BUT THIS DOES NOT MATCH THE KNOWN KEYSTREAM WHERE $S_{n+5} \equiv 0$. SO $m \neq 2$.

- IF $m=3$, WE CAN WRITE

$$\begin{aligned} S_{n+3} &= a_0 S_n + a_1 S_{n+1} + a_2 S_{n+2} \\ S_{n+4} &= a_0 S_{n+1} + a_1 S_{n+2} + a_2 S_{n+3} \\ S_{n+5} &= a_0 S_{n+2} + a_1 S_{n+3} + a_2 S_{n+4} \end{aligned}$$

SO THAT WE GET

$$\begin{aligned} 0 &\equiv a_1 + a_2 \\ 1 &\equiv a_0 + a_1 \\ 0 &\equiv a_0 + a_2 \end{aligned}$$

WHICH HAS NO SOLUTION, THUS $m \neq 3$

- IF $m=4$ WE WRITE 4 EQUATIONS WITH $S_n, \dots, S_{n+7}$. IN THIS CASE, WE OBTAIN THE SYSTEM

$$1 \equiv a_1 + a_2$$
$$0 \equiv a_0 + a_1 + a_3$$
$$1 \equiv a_0 + a_2$$
$$1 \equiv a_1 + a_3$$

WHICH HAS SOLUTION $(a_0\ a_1\ a_2\ a_3) = (1\ 1\ 0\ 0)$. WE CAN CHECK THAT THIS LFSR DOES INDEED GENERATE ALL OF THE KNOWN KEYSTREAM, AND IT IS MOST PROBABLY THE CORRECT SOLUTION.

- AN IMORTANT LESSON IS THAT LINEARITY IS VERY BAD FOR CRYPTOGRAPHY. LINEAR FUNCTIONS BEHAVE IN A PREDICTABLE WAY. THIS MAKES IT EASY TO ANALYZE THEM (WHICH IS WHY E.G. LINEAR ALGEBRA IS SO WELL DEVELOPED WHILE ALMOST NOTHING CAN BE SAID ABOUT NON-LINEAR FUNCTIONS) BUT THIS IS PRECISELY WHAT ONE WISHES TO AVOID IN CRYPTOGRAPHY.

- LFSR'S ARE STILL USEFUL IN CRYPTOGRAPHY AND IN FACT BLOCK CIPHERS ALSO CONTAIN A LOT OF LINEAR COMPONENTS. THE CORRECT APPROACH IS TO COMBINE LFSR'S INTO MORE COMPLICATED SYSTEMS AND TO POSSIBLY USE THEM ALONGSIDE NON-LINEAR COMPONENTS.

- EXAMPLE: THE TRIVIUM STREAM CIPHER (SEE SLIDES) COMBINES THREE LFSR'S IN A NON-LINEAR MANNER. AS OF TODAY, NO EFFICIENT ATTACK AGAINST TRIVIUM IS KNOWN.

- LFSR'S CAN BE GENERALIZED TO NFSR'S (NON-LINEAR FSR'S) WHICH ARE MORE SECURE PER SE BUT ARE LESS WELL UNDERSTOOD. IN AN NFSR, THE NEXT BIT s_{n+m} IS COMPUTED AS

$$s_{n+m} = f(s_n, s_{n+1}, \ldots, s_{n+m-1})$$

FOR A NON-LINEAR FUNCTION $f : \mathbb{Z}_2^m \to \mathbb{Z}_2$.

HOMEWORK PROBLEMS

① FOR EACH PAIR (a, b) DECIDE WHETHER IT IS A VALID KEY FOR THE AFFINE CIPHER. IF SO ENCRYPT "HELLO WORLD" WITH IT THEN DECRYPT IT AND COMPARE AGAINST THE ORIGINAL PLAINTEXT.
- $(17, 3)$ · $(3, 17)$ · $(23, 17)$ · $(0, 8)$ · $(8, 0)$

② GENERALIZED AFF. CIPHER: INSTEAD OF WORKING WITH LETTERS WE WORK WITH NUMBERS, AND INSTEAD OF $\mathbb{Z}_{26}$ WE TAKE $\mathbb{Z}_{31}$. REPEAT ① WITH THE PLAINTEXT 30 − 7 − 5 − 1 − 21.

③ COMPUTE THE GCD AND BEZOUT COEFFICIENTS OF $(100345, 25025)$ AND $(7208, 7869)$

④ HOW MANY ELEMENTS OF $\mathbb{Z}_n$ ARE INVERTIBLE FOR:
- $n = 30$ · $n = 37$ · $n = 64$ · $n = 9677$

FIND THE INVERSE OF 5 FOR $n = 37$ AND $n = 64$.

⑤ FIND $x, y \in \mathbb{Z}$ SOLVING $17x + 101y = 1$.

* ⑥ IF $a \cdot b \equiv 0 \mod p$ FOR p PRIME, SHOW THAT AT LEAST ONE OF a AND b IS 0 MODULO p.

* ⑦ SHOW THAT IF $a, b, n \in \mathbb{Z}$ SUCH THAT $GCD(a, n) = 1$ AND $n | ab$, THEN $n | b$. IS THIS TRUE IF $GCD(a, n) \neq 1$?

* ⑧ FOR $p \geq 3$ PRIME, SHOW THAT $x \equiv \pm 1 \mod p$ ARE THE ONLY SOLUTIONS TO $x^2 \equiv 1 \mod p$.

⑨ GIVEN THE CIPHERTEXT C = 011010111001, FIND A KEY FOR WHICH C DECRYPTS TO EACH OF THE FOLLOWING PLAINTEXTS USING THE VERNAM CIPHER:
- P = 0011 0011 0011
- P = 0101 0000 1111
- P = 01100 111 1101

⑩ FOR THE LFSR'S REPRESENTED BY THE FOLLOWING POLYNOMIALS, FIND OUT HOW LONG IT TAKES FOR THE LFSR TO LOOP DEPENDING ON ITS INITIAL STATE, I.E. PARTITION ITS STATE SPACE INTO CYCLES:
- $p(x) = x^5 + x^3 + x + 1$
- $p(x) = x^5 + x^2 + 1$
- $p(x) = x^5 + x^3 + x^2 + x + 1$

$\boxed{*}$ ⑪ IMPLEMENT AN LFSR AS A PROGRAM: GIVEN AN INITIAL STATE E.G. 001101 AND A NUMBER k OF ITERATIONS, THE PROGRAM SHOULD OUTPUT A KEYSTREAM OF LENGTH k.