COMP3260
Data Security

Lecture 3

Prof Ljiljana Brankovic

Lecture Overview

1. Galois fields GF(p) and GF(2n)
2. Entropy
3. Theoretical Secrecy
4. Rate of the Language
5. Redundancy
6. Equivocation
7. Perfect Secrecy
8. One-Time Pad

Number Theory and Finite Fields

- Text: Chapter 5 Finite Fields
- "Cryptography and Data Security" by D. Denning [2]

Note that in-text references and quotes are omitted for clarity of the slides. When you write as essay or a report it is very important that you use both in-text references and quotes where appropriate.

Groups, Rings and Fields

Group

(A1) Closure under addition:	If a and b belong to S, then $a + b$ is also in S
(A2) Associativity of addition:	$a + (b + c) = (a + b) + c$ for all a, b, c in S
(A3) Additive identity:	There is an element 0 in R such that $a + 0 = 0 + a = a$ for all a in S
(A4) Additive inverse:	For each a in S there is an element $-a$ in S such that $a + (-a) = (-a) + a = 0$

Abelian Group

(A5) Commutativity of addition:	$a + b = b + a$ for all a, b in S

Ring

(M1) Closure under multiplication:	If a and b belong to S, then ab is also in S
(M2) Associativity of multiplication:	$a(bc) = (ab)c$ for all a, b, c in S
(M3) Distributive laws:	$a(b + c) = ab + ac$ for all a, b, c in S $(a + b)c = ac + bc$ for all a, b, c in S

Commutative Ring

(M4) Commutativity of multiplication:	$ab = ba$ for all a, b in S

Integral Domain

(M5) Multiplicative identity:	There is an element 1 in S such that $a1 = 1a = a$ for all a in S
(M6) No zero divisors:	If a, b in S and $ab = 0$, then either $a = 0$ or $b = 0$

Field

(M7) Multiplicative inverse:	If a belongs to S and $a \neq 0$, there is an element a^{-1} in S such that $aa^{-1} = a^{-1}a = 1$

Galois fields GF(p) and GF(2^n)

- Real and integer arithmetic is not suitable for cryptography because information is lost through rounding or truncation.

- Also, for efficiency, we prefer to work with integers that have a given number of bits. For that reason we prefer to use modular arithmetic – integers modulo n.

- If n is a prime number, we have a finite field of order p, which is referred to as Galois field, in honour of Évariste Galois, a French mathematician who first studied them.

Galois fields GF(p) and GF(2^n)

- Many recent ciphers are based on arithmetic in a Galois field *GF(p)*, where *p* is a prime number.

- Note that because *p* is a prime, every integer $a \in [1,p-1]$ is relatively prime to *p* and thus has an inverse modulo *p*.

- The set of integers *mod p* together with binary operations ‘addition’ and ‘multiplication’ is called a field: it is an integral domain where every element besides 0 has a multiplicative inverse.

- That means that we can do addition, subtraction, multiplication and division without leaving the set.

Galois fields GF(p) and GF(2^n)

- Another type of Galois field used in cryptography is $GF(q^n)$ where elements are polynomials of degree $n-1$ of the form

  $$a = a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

  where the coefficients a_i are integers *mod q*

- Each element a is a residue *mod* $p(x)$ where $p(x)$ is an irreducible polynomial of degree n. (Irreducible means that p cannot be factored into polynomials of degree less than n)

Galois fields GF(p) and GF(2^n)

- We are particularly interested in the fields *GF(2^n)* where the coefficients are binary digits *0* and *1*. Computing in *GF(2^n)* is very efficient and addition and subtraction correspond to $\oplus$ (exclusive-or) of the coefficients. Multiplication can also be done efficiently, but requires dividing by *p(x)*.

- Advanced Encryption Standard (AES) uses arithmetic in the finite field *GF(2^8)* with the irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$. Every element *a* of this field can be represented as bit vector of length *8*, e.g., *11011011*.

Galois fields $GF(p)$ and $GF(2^n)$

- Computing in *$GF(2^n)$* is more efficient than computing in *$GF(p)$*, where $2^{n-1} < p < 2^n$. Why?
- <u>Space efficient</u>: All *n*-bit vectors correspond to elements of *$GF(2^n)$*, which is not the case for *$GF(p)$*.
- <u>Time efficient</u>: Arithmetic is more efficient in *$GF(2^n)$* than in *$GF(p)$*.

Galois fields GF(p) and GF(2^n)

- Multiplicative inverses:

  Note that in *GF(2^n)*, *p(x)* is irreducible, so it is relatively prime to all polynomials of order *n-1*, but we are not counting the polynomial whose coefficients are 0 (0-vector).

Thus:

$$\phi(p(x)) = 2^n-1$$

$$a^{-1} = a^{\phi(p(x))-1} \bmod p(x) = a^{(2^n)-2} \bmod p(x)$$

Alternatively

$a^{-1} = inv(a, p(x))$ (Using Euclid's algorithm).

Galois fields GF(p) and GF(2^n)

1. Let a = 10101 and b = 01100. In GF(2^5), find c = a+b.

   Solution:

   a = 1 0 1 0 1
   $\oplus$ b = 0 1 1 0 0
   ----
   c = 1 1 0 0 1

2. Let a = 10101 and b = 01100. In GF(31), find c = a+b.

   Solution:

   a = 1 0 1 0 1 (21)
   b = 0 1 1 0 0 (12)
   c = 1 0 0 0 0 1 (33)

   33 mod 31 = 2 hence c = 0 0 0 1 0 (2)

Galois fields GF(p) and GF(2^n)

3. Let a = 10111001 and b = 01101110. In GF(2^8), find c = a-b.

   Solution: c = 11010111

4. Let a = 1101 and b=0101. In GF(2^4), find c = a+b.

   Solution: c = 1000

Galois fields GF(p) and GF(2^n)

5. Let a = 101. In GF(2^3) with irreducible polynomial $p(x) = x^3 + x + 1$ find d=a*a.

Solution: Multiply a * a

```
   101
   101
   ---
   101
  000
 101
-------
 10001
```

		1	0	1
		1	0	1
		1	0	1
	0	0	0	
1	0	1		
1	0	0	0	1

Divide by p(x) = 1011:

```
     _____
1011)10001
      1011
    ------
       111 = d
```

Galois fields GF(p) and GF(2)

6. Let a = 111 and b = 100. In $GF(2^3)$ with irreducible polynomial $p(x) = x^3+x+1$ find d=a*b

Solution: Multiply a*b:

```
  1 11
   100
   ---
   000
  000
1 11
 --------
 1 1100
```

Divide by p(x) = 1011:

```
1011 ) 11100
       1011
        ------
        1010
        1011
        ------
        0001 = d      d=001
```

Galois fields GF(p) and GF(2^n)

7. Let a=100. If GF(2^3) with irreducible polynomial $p(x)= x^3 + x + 1$ find a^{-1} and verify that $a \times a^{-1} \bmod p(x)=1$

Solution:

$\phi(p(x)) = 2^3-1 = 7$, $a^{-1} = 100^6 \bmod 1011$

$100^2 = 100 * 100$

```
      100
      100
      -----
      000
     000
    100
   ---------
   10000
```

Divide by p(x) = 1011:

```
1011 ) 10000
       1011
       -------
         110
```

Hence $100^2 = 110$

Galois fields GF(p) and GF(2^n)

100^4 = 110 * 110

```
    110
    110
    ---
    000
   110
  110
  --------
  10100
```

Divide by p(x) = 1011:

```
         _______
1011 ) 10100
       1011
       --------
          010
```

Hence 100^4 = 010

Galois fields GF(p) and GF(2^n)

$100^6 = 100^2 * 100^4 = 110 * 010$

```
   110
   010
   ---
   000
  110
000
--------
01100
```

Divide by p(x) = 1011:

```
       ____
1011 ) 1100
       1011
       --------
        111
```

Hence $100^6 = 111$

$a^{-1} = 100^6 \bmod 1011 = 111$

ENTROPY continued
Example 2

The messages describe penalty shoot-outs; we have g=goal and m=miss, with probabilities

$p(g)=0.9$ and

$p(m)=0.1$.

$$H(X) = -(0.9 \times \log_2(0.9) + 0.1 \times \log_2(0.1))$$

$$= -(-0.13680 - 0.3321) = 0.469$$

Example 3

X is one of the k characters $c_1, c_2, \ldots, c_k$, where each character has probability 1/k.

$$H(X) = -\sum_{1}^{k} \left(\frac{1}{k}\right) \times \log_2\left(\frac{1}{k}\right) = \log_2 k$$

If k=256 then H(X)=8.

That is, if every character in the set of 256 characters has the same probability then the average number of bits per character is 8.

In general, for $n=2^k$, H(X)=k, that is, k bits are needed to encode each possible message.

Example 4

Suppose $n=1$ and $p(X)=1$. What is the entropy of the message?

$$H(X) = \log_2 1 = 0$$

There is no information in this message because there is no choice.

Entropy - Cont.

Given n messages, H(X) is maximal for

$$p(X_1) = p(X_2) = \ldots = p(X_n) = 1/n,$$

that is, when all messages are equally likely.

H(X) decreases as the distribution of messages becomes more and more skewed, reaching a minimum of $H(X)=0$ when $p(X_i)=1$ for some message X_i.

The entropy of a message measures its **uncertainty**: it gives the number of bits of information that must be learned when the message has been distorted by a noisy channel or hidden in ciphertext.

Theoretical Secrecy

Information theory provides a theoretical foundation for cryptography.

Information theory measures the theoretical **secrecy** of a cipher by the uncertainty about a plaintext given the corresponding ciphertext.

Perfect secrecy is achieved if no matter how much ciphertext is intercepted, nothing can be learned about the plaintext.

The only perfectly secret cipher is **one-time pad**.

Theoretical Secrecy

All other ciphers leave some information about the plaintext in the ciphertext.

As the length of the ciphertext increases, the uncertainty about the plaintext usually decreases, eventually reaching 0. At that point there is enough information to determine the plaintext uniquely and the cipher is breakable (at least in theory).

Most ciphers are **theoretically** breakable with only a few characters of ciphertext.

Theoretical Secrecy

This does not necessarily mean that the ciphers are insecure: the computational requirements needed to determine the plaintext may exceed available resources.

The important question is not whether a cipher is **unconditionally secure** but whether it is **computationally (practically) secure**.

Rate of the Language

Consider a language ***L*** consisting of messages of N characters.

The **rate of the language** is the average entropy per character, that is,

$$r = H(X) / N$$

Real languages consist of messages of varying lengths.

In that case we can define the rate of the language for messages of length N, say r_N.

Rate of the Language

As N increases, r_N tends to a constant r which is then the **rate of the language**.

For large N, estimates of r for English range from 1.0 bit/letter to 1.5 bit/letter.

We shall use **1.5 bit/letter** as the estimate for English in our calculations.

Absolute Rate of the Language

If the alphabet of L consists of L characters then the absolute rate R of the language is $R = \log_2 L$.

The absolute rate is the maximum entropy of the characters under any probability distribution.

That is, the absolute rate is the maximum number of bits of information that could be encoded in each character assuming all possible sequences of characters are equally likely.

Absolute Rate of the Language

If all sequences of characters in a language have the same probability then $r = R$.

What is the absolute rate of English?

The absolute rate of English is $R = \log_2 26 = 4.7$ bits per letter.

Redundancy

The absolute rate of English is significantly greater that the actual rate because English is highly redundant.

Mst ids cn b xpresd n fwr ltrs, bt th xprnc s mst nplsnt!

In any natural language, as well as in programming languages, redundancy arises from the structure of the language.

Redundancy

The redundancy is reflected in the statistical properties of actual meaningful messages:

- single letter frequency distribution
- diagram frequency distribution
- trigram frequency distribution
- N-gram frequency distribution

As longer sequences are considered, the proportion of meaningful messages to the total number of possible letter sequences decreases.

Redundancy

In practice, the rate of language (entropy per character) is determined by estimating the entropy of N-grams for increasing values of N.

As N increases, the entropy per character decreases because there are fewer choices and some choices are much more likely than others.

The rate of language is estimated by extrapolating for large N.

Redundancy

The **redundancy D** of a language with rate r and absolute rate R is defined as

$$D = R - r.$$

For English,

$$D = 4.7 - 1.5 = 3.2$$

Thus, English is 68% redundant, since D/R = 0.68.

When using the rate of 1 it is around 79% redundant.

Equivocation

The uncertainty of a message can be further reduced given additional information.

Example: Suppose X is a 32-bit integer, all values equally likely so H(X) = 32. Suppose we learn that X is even. How much does this additional information reduce the entropy of X?

By 1 bit because all even integers have 0 as their last bit.

Equivocation

The entropy of a message X, given some additional information Y, is measured by the **equivocation** $H_Y(X)$, the uncertainty about X given knowledge of Y.

The **equivocation** $H_Y(X)$ is the **conditional entropy** of X given Y:

$$H_Y(X) = -\sum_{X,Y} p(X,Y)\log_2 p_Y(X) =$$

$$= \sum_{X,Y} p(X,Y)\log_2 \frac{1}{p_Y(X)} =$$

$$= \sum_{Y} p(Y)\sum_{X} p_Y(X)\log_2 \frac{1}{p_Y(X)}$$

Equivocation

$p_Y(X)$ is the conditional probability of message X given message Y and p(X,Y) is the joint probability of message X and message Y:

$$p(X,Y) = p_Y(X)\,p(Y)$$

If events X and Y are independent, then $p_Y(X)=p(X)$

And we have

$$p(X,Y) = p(X)\,p(Y)$$

and

$$H_Y(X) = H(X)\sum_Y p(Y) = H(X)$$

Equivocation

Example: Let $n=4$ and $p(X)=1/4$ for each message X so $H(X)=\log_2 4=2$. Let $m=4$ and $p(Y)=1/4$ for each message Y. Suppose each message Y narrows down the choice of X to two of the four messages, both equally likely:

Y_1: X_1 or X_2 $\quad$ Y_2: X_2 or X_3
Y_3: X_3 or X_4 $\quad$ Y_4: X_4 or X_1

What is the equivocation of X given Y?

In this case the knowledge of Y reduces the uncertainty of X to 1 bit.

Perfect Secrecy

Shannon studied the information theoretic properties of cryptographic systems in terms of three classes of information:

- plaintext messages M occurring with known probabilities $p(M)$, where $\Sigma_M\, p(M)=1$;
- ciphertext messages C occurring with known probabilities $p(C)$, where $\Sigma_C\, p(C)=1$;
- keys K chosen with probabilities $p(K)$, where $\Sigma_K\, p(K)=1$.

Perfect secrecy is defined by the condition

$$p_C(M) = p(M)$$

where $p_C(M)$ is the probability that M was sent given that C was received.

Perfect Secrecy

The probability of receiving C given that M was sent is the sum of the probabilities $p(K)$ of the keys K that encipher M as C:

$$p_M(C) = \sum_{K, E_K(M)=C} p(K)$$

A necessary and sufficient condition for perfect secrecy is that for every C and for all M $p_M(C)=p(C)$.

Perfect Secrecy

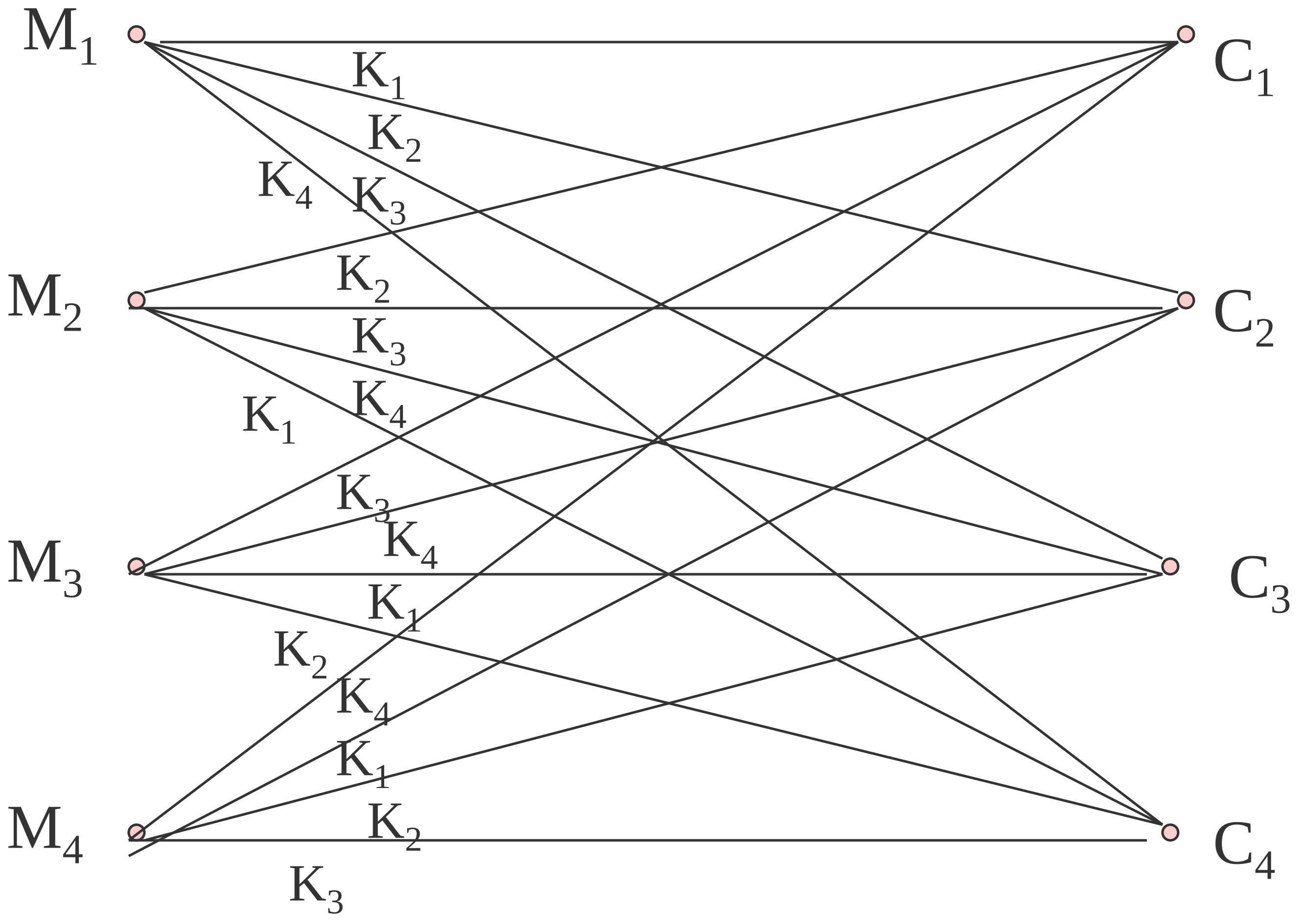
So the probability of receiving a particular ciphertext C given that M was sent is the same as the probability of receiving C given that some other (any other) message M was sent.

Perfect secrecy is possible using a completely random key at least as long as the message it enciphers.

Perfect secrecy requires that the number of keys must be at least as great as the number of possible messages.

Otherwise there would be some message M such that for a given C, no K deciphers C into M , implying that $p_C(M)=0$. However, M is not impossible, so $p_C(M) \neq p(M)$.

Perfect Secrecy



Perfect Secrecy

Example: Suppose we intercept a ciphertext which was produced by a Caesar cipher, with key K.

C=DOXDRYECKXNWOXGYEVNXYDLOOXYEQR

Is this cipher perfectly secure?

No. We cannot achieve perfect secrecy because the number of possible keys is smaller than the number of possible English sentences of length 31. This cipher is easily broken because only one of the possible 26 keys (K=10) produces a meaningful message: **TENTHOUSANDMENWOULDNOTBEENOUGH**

Perfect Secrecy

We have $p_C(M)=1$ and $p_C(M')=0$ for every other message M'.

$p_M(C)=p(10)=1/26$ and $p_M{}'(C)=0$ for every other message M'.

$p_C(M)$ is certainly greater than $p(M)$ and $p_M(C)$ is greater that $p(C)$.

One-Time Pad

Modification of the preceding example to achieve perfect secrecy: shift each letter not by a constant number of places but by a random number.

Then $K=k_1k_2\ldots$, where each k_i is a random integer in the range [0,25].

Perfect secrecy is achieved since any 31 character long message could be enciphered to C.

One-Time Pad

The ciphertext in the last example could have resulted from using the key K =3,3,4,22,3,4,24,21,22,10,9,10,14,10,20,16,24,14,10,14,11,18, 20,18,19,22,14,13
to encrypt the message

M′=**ALTHOUGHONEMANMIGHTJUSTSUFFICE**

A cipher that uses a non-repeating random key stream is called a **one-time pad**.

One time pads are the only ciphers that achieve perfect secrecy.

Next Week

1. Unicity Distance
2. Symmetric Cipher Model
3. Kerckhoffs' Laws
4. Codes and Ciphers
5. Classical ciphers
    1. Transposition Ciphers and How to Break Them
    2. Substitution Ciphers and How to Break Them

- Text Chapter 2
- "Cryptography and Data Security" by D. Denning - Information theory

References

1. W. Stallings. “Cryptography and Network Security”, Pearson, global edition, 2016.

2. D. Denning. “Cryptography and Data Security”, Addison Wesley, 1982.