Information theory, PRNG, and Symmetric Cipher Properties and Attacks

CSE 468 Fall 2024

jedimaestro@asu.edu

What do we need randomness for?

- Private keys
- Session keys
- Initialization Vectors
- Random padding
- ...

Don’t do this...

```
int i = 10000000 + new Random().nextInt(89999999);
int j = 10000000 + new Random().nextInt(89999999);
return (String.valueOf(i) + String.valueOf(j)).getBytes();
```

Figure 1: Decompiled Java method generating an AES session key in version 6.3.0.1920.

```
Random random = new Random(System.currentTimeMillis());
byte[] bArr = new byte[8];
byte[] bArr2 = new byte[8];
random.nextBytes(bArr);
random.nextBytes(bArr2);
return new SecretKeySpec(ByteUtils.mergeByteData(bArr, bArr2), "AES");
```

Figure 2: Decompiled Java method generating an AES session key in version 6.5.0.2170.

https://arxiv.org/pdf/1802.03367.pdf

How do we measure “information”?

- Entropy
  - Don’t be confused if you’ve heard this term in a physics class
    - Entropy in physics is the information we don’t have about energy, which leads to wasted energy
    - Entropy in information theory is a measure of how surprised we’ll be when we learn information, which leads to useful information

Requirements (Shannon, 1948)

1) $I(p) \geq 0$ (information is non-negative, $p \geq 1$)

2) $I(1) = 0$ (events that always occur carry no information)

3) $I(p1\ p2) = I(p1) + I(p2)$ (information due to independent events is additive)

Also, continuity, symmetry, and maximum when all possible events are equiprobable.

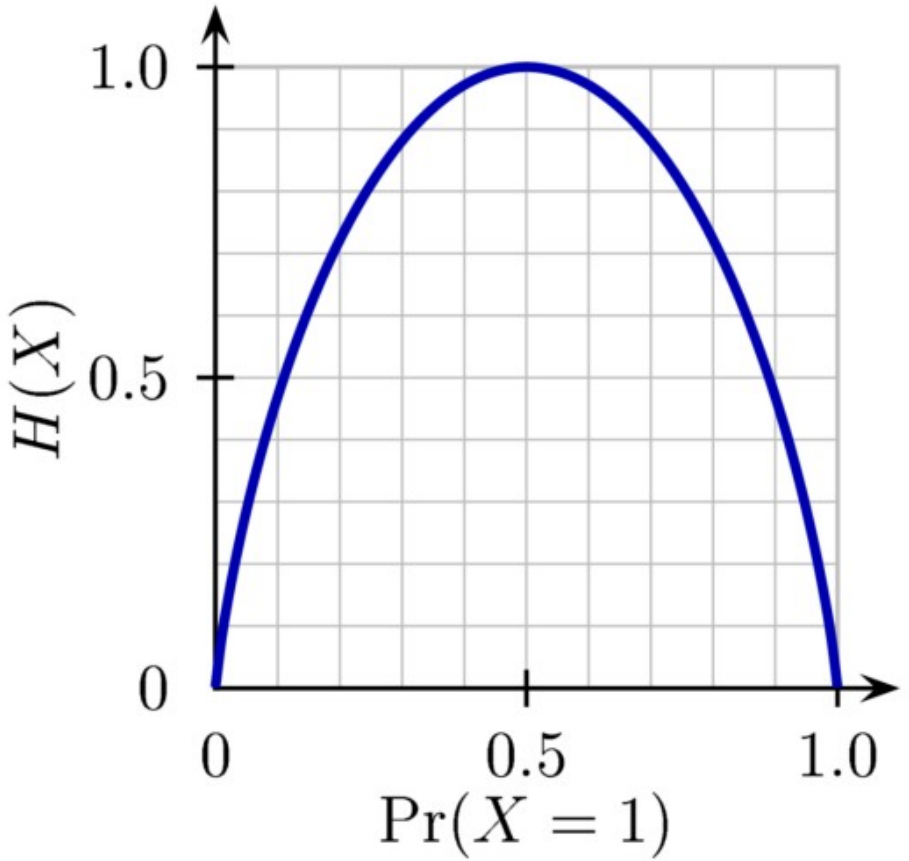
$$I(p) = \log(1/p)$$

1) $I(1/2) = 0.30102999566...$
2) $I(1) = 0$
3) $I(1/2) + I(1/3) = \log(2) + \log(3) = 0.77815125038...$
Joint probability: $I(1/6) = 0.77815125038...$

Continuity: $I(1/2.01) = 0.30319605742...$
Symmetry: $\log(3) + \log(2) = 0.77815125038...$
Maximum: $\log 3 + \log 3 + \log 3 = 1.43136376416...$
$\log 2 + \log 4 + \log 4 = 1.20411998266...$

Information = Entropy = Surprise

$$H[p] = -\sum_{i=1}^{k} p_i \log p_i$$



Side note – Differential Entropy

https://en.wikipedia.org/wiki/Differential_entropy

$$h(X) = \mathrm{E}[-\log(f(X))] = -\int_{\mathcal{X}} f(x)\log f(x)\,dx$$

f(x) is a probability density function for the signal, the more the signal “jumps around” the higher the entropy, therefore modulating higher frequencies means more entropy and therefore more bandwidth.

Not a pop quiz #1

- When a 3yo walks by with a stepstool…
  - 4 times out of 10 it's to get something they're not supposed to have
  - 2 times out of 10 it's to climb up to somewhere they're not supposed to be
  - 1 time out of 10 it's to wash their hands
  - 1 time out of 10 it's to get something they're allowed to have
  - 1 time out of 10 it's to use as a dollhouse
  - 1 time out of 10 it's to turn over and use as a storage bin
- What is the entropy of each instance of 3yo stepstool habits?

Answer

Input:

$$-0.4 \log_2(0.4) - 0.2 \log_2(0.2) - 4\,(0.1 \log_2(0.1))$$

Result:

$$2.32193\ldots$$

Not a pop quiz #2

- There are three possible states the Tempe weather could be in during any given hour on a summer day (very hot and bright out, very hot and it’s nighttime, monsoonal rains). What probability distribution over these events would give the maximum entropy in terms of what you might observe in a randomly chosen hour from the summer?

Answer

Input:

$$-\frac{1}{3}\log_2\left(\frac{1}{3}\right)-\frac{1}{3}\log_2\left(\frac{1}{3}\right)-\frac{1}{3}\log_2\left(\frac{1}{3}\right)$$

Exact result:

$$\frac{\log(3)}{\log(2)}$$

$\log(x)$ is the natural logarithm

Decimal approximation:

More digits

1.584962500721156181453738943947816508759814407692481060455...

Not a pop quiz #3

You have 12 coins, one is counterfeit. The counterfeit is either slightly heavier or slightly lighter, otherwise it's impossible to tell. You have a balance. Using the balance the fewest number of times, find the counterfeit coin.

Not a pop quiz #4

- A measure of the information of a **random process**
- Pop quiz #3: Based on the above definition, order the following binary sequences from most entropy to least entropy?:

  A) 11111111000000000000000011111111

  B) 10111001110011001100010000110100

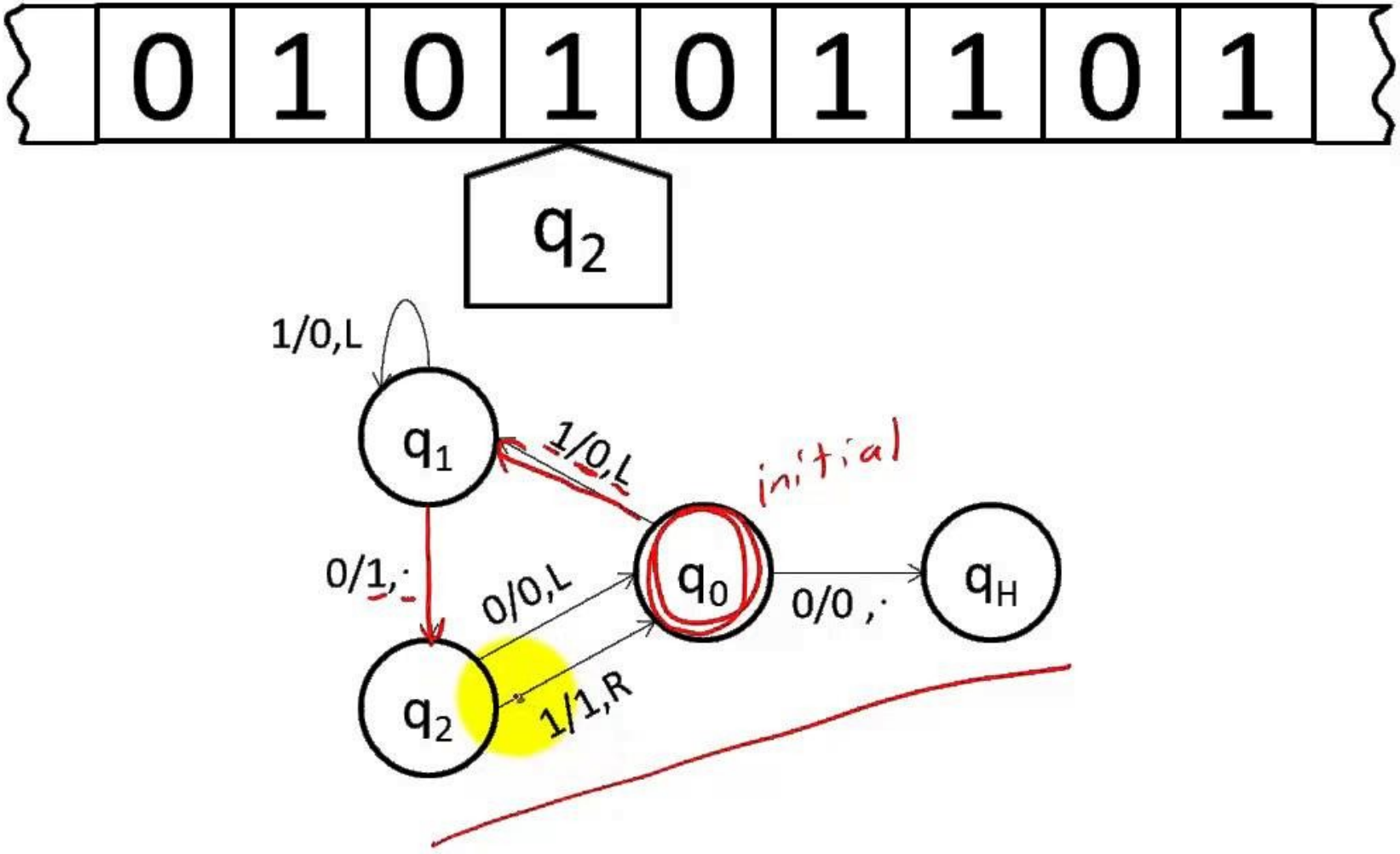
  C) 00000000000000000000000000000000

  D) 00010010000000100000000100000001

I pity the fool who uses the word “entropy” to describe a bit sequence or string without realizing that they are implicitly talking about algorithmic entropy (*a.k.a.*, Kolmogorov complexity) rather than the standard definition of entropy that Claude Shannon used to describe random processes!

Turing machines

Source: https://www.youtube.com/watch?v=gJQTFhkhwPA



Halting problem

- From a description of an arbitrary computer program and an input, determine whether the program will finish running or continue to run forever.

  P: if (halts(P)) then: while true {}; else halt;

Gödel's Incompleteness Theorem

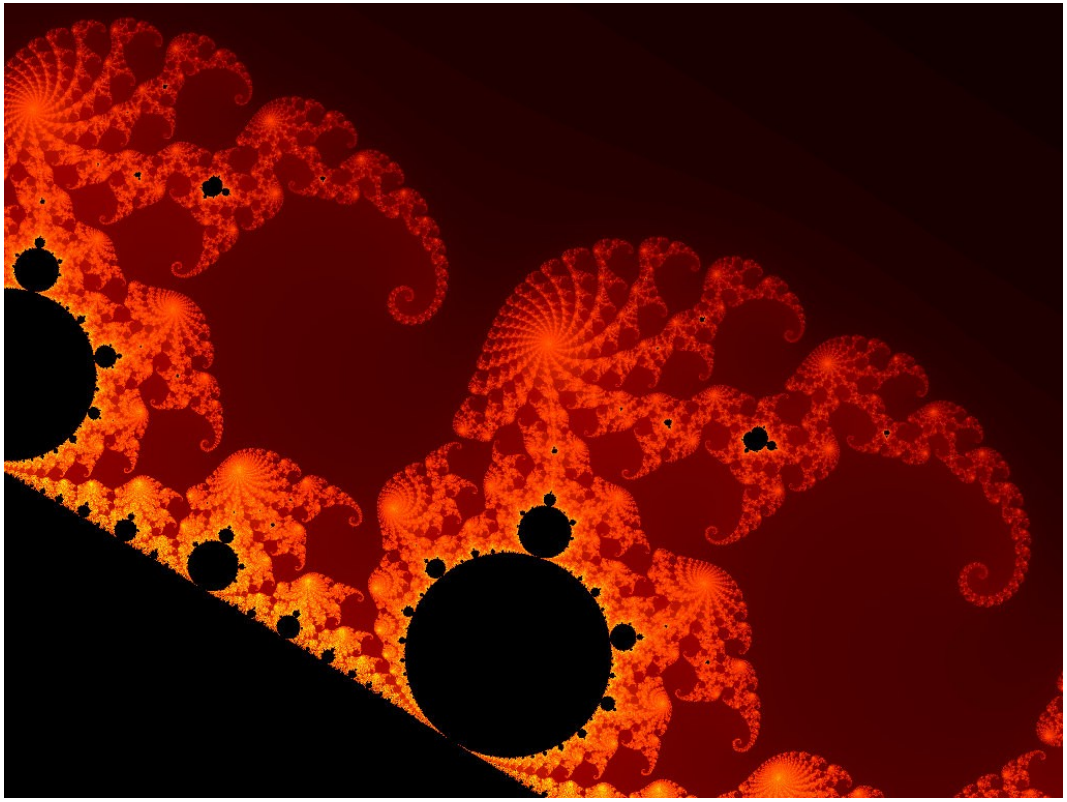
- In axiomatic systems capable of arithmetic, can't separate mathematics and meta-mathematics, can always form sentences of the form:

  This sentence is false.

- As Einstein put it:
  - “As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality.”
  - “We can't solve problems by using the same kind of thinking we used when we created them.”

Kolmogorov complexity

- Defined as the length of the shortest computer program that produces the object as output.



Chaitin's incompleteness theorem

- “The fact that a specific string is complex cannot be formally proven, if the complexity of the string is above a certain threshold.” [Wikipedia]
- Berry's paradox:
  - “the smallest positive integer not definable in fewer than twelve words”

PRNG

- Example:

  $X_{i+1} = X_i * a + b \bmod m$

- Seed
- Period
- Can be made cryptographically strong
  - Attacker knows the algorithm and lots of past bits
  - *E.g.*, take the above and encrypt it with AES and a well chosen key.
  - For more detailed examples, look up Yarrow or Fortuna.

PRNG problems

- Seeding problems
  - E.g., when you first boot an embedded device
- Periodicity problems
  - Never use srand(time) and rand() for crypto
- Backdoors
  - Dual_EC_DRBG (see Snowden revelations)
- Predictability
  - Witty worm

So, where do random numbers for crypto come from in practice on everyday user devices?

Linux kernel's entropy pool

Secure hash function (e.g., SHA1 or ChaCha20-based)

`/dev/urandom`
`/dev/random`

What makes a good symmetric crypto algorithm?

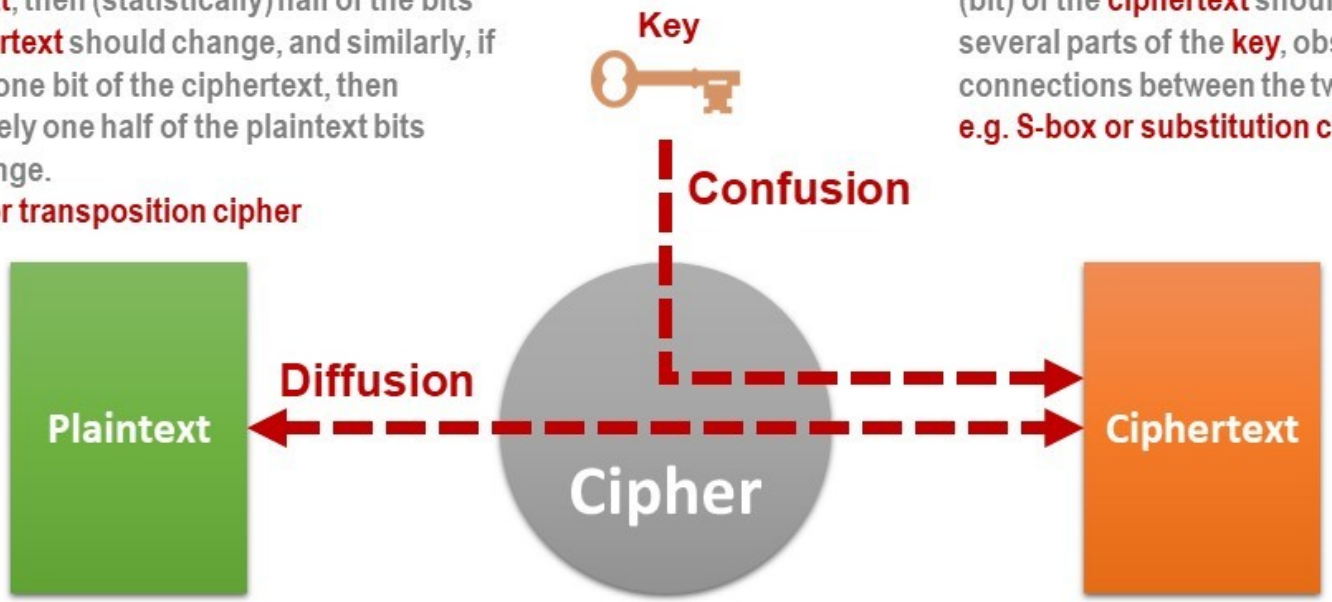
Lots of things, but two you should know are confusion and diffusion (diffusion is also known as the avalanche effect).

Claude Shannon, *A Mathematical Theory of Cryptography (1945 classified report)*

Confusion and Diffusion

Diffusion means that if we change a single bit of the **plaintext**, then (statistically) half of the bits in the **ciphertext** should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change.
e.g. P-box or transposition cipher

Confusion means that each binary digit (bit) of the **ciphertext** should depend on several parts of the **key**, obscuring the connections between the two.
e.g. S-box or substitution cipher



https://en.wikipedia.org/wiki/Confusion_and_diffusion

三誠業 Amicliens

https://wentzwu.com/2019/09/02/confusion-and-diffusion/

Attacks on block ciphers

- Linear and differential cryptanalysis
  - NSA must have known about these when giving input about DES, rest of the world found out in the 1990s
- Many others
  - E.g., rotational cryptanalysis
- CBC padding oracle attacks and others that are typically performed on live systems

For more details and the image source for the following two slides, see:
A Tutorial on Linear and Differential Cryptanalysis, by Howard M. Heys
https://jedcrandall.github.io/courses/cse539spring2023/ldc_tutorial.pdf

Linear cryptanalysis

- Solve for the key using plaintext/ciphertext pairs and linear approximations
- XOR is linear arithmetic modulo 2, permutations are also linear, only S-boxes save you

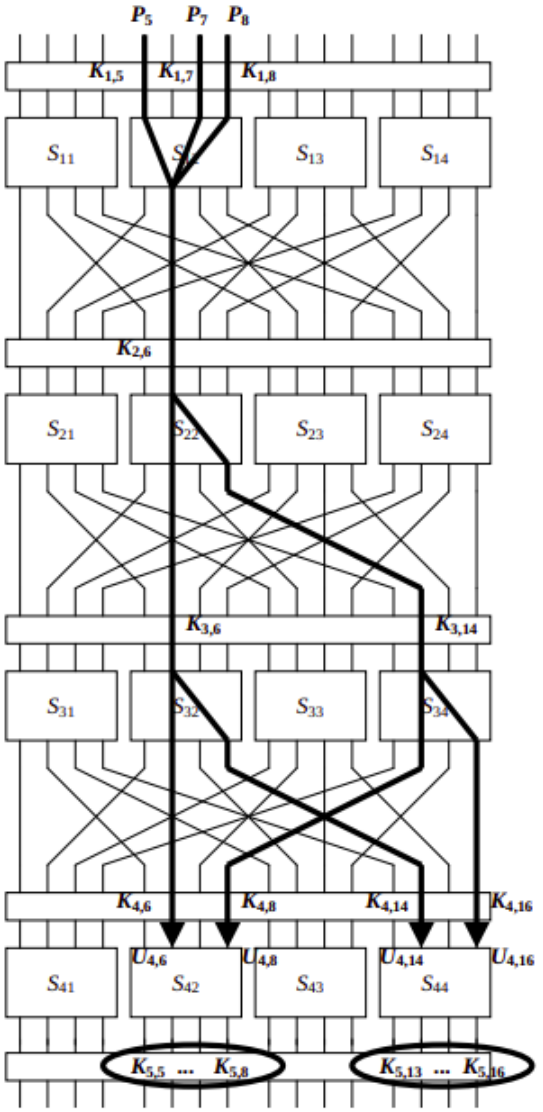


Figure 3. Sample Linear Approximation

Differential cryptanalysis

- Solve for key using plaintext/ciphertext pairs and propagated bit differences
- XOR and permutations don’t hide bit differences, only the S-boxes save you

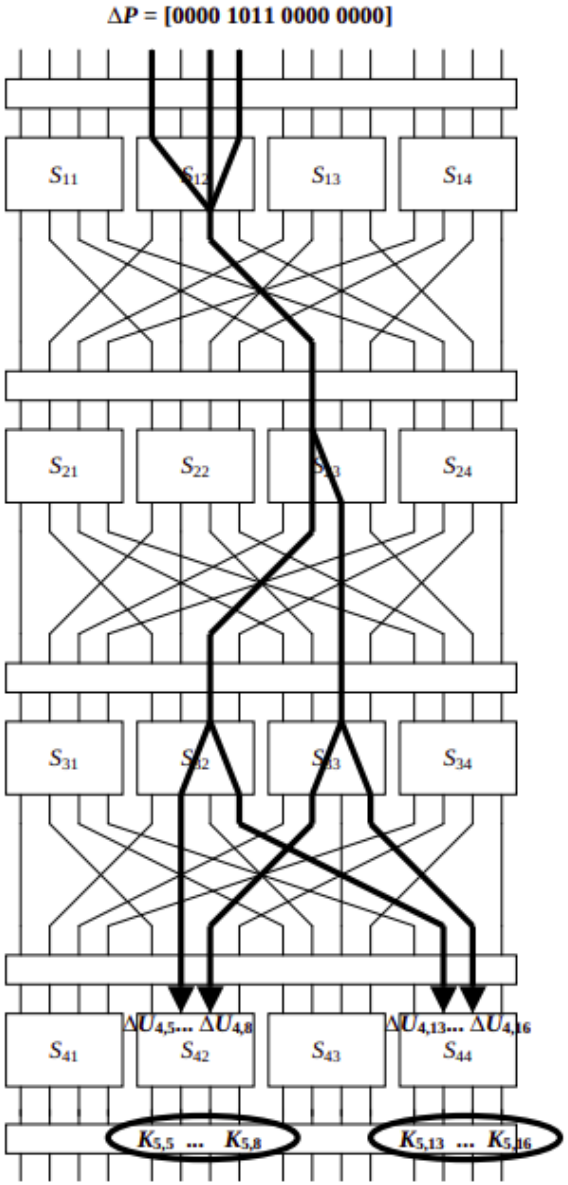


Figure 5. Sample Differential Characteristic

AES S-box requirements (review)

- Can’t pull it out of our &%# like the NSA did for DES
- Should have good nonlinear properties
- Should be reversible
  - Don’t want to use a Feistel structure for performance reasons

Galois fields...
https://www.youtube.com/watch?v=Ct2fyigNgPY

Sources

- https://en.wikipedia.org/wiki/Entropy_%28information_theory%29
- http://www.amazon.com/Elements-Information-Theory-Telecommunications-Processing/dp/0471241954/ref=sr_1_8?ie=UTF8&qid=1457368787&sr=8-8&keywords=information+theory
- http://www.amazon.com/Mathematical-Theory-Communication-Claude-Shannon/dp/0252725484/ref=sr_1_1?s=books&ie=UTF8&qid=1457368808&sr=1-1&keywords=information+theory
- https://en.wikipedia.org/wiki/Turing_machine
- https://www.youtube.com/watch?v=gJQTFhkhwPA
- https://en.wikipedia.org/wiki/Halting_problem
- http://www.icir.org/vern/papers/witty-imc05.pdf