CSCI-GA.3205 Applied Cryptography & Network Security

Department of Computer Science
New York University

PRESENTED BY DR. MAZDAK ZAMANI
mazdak.zamani@NYU.edu

Affine Cipher Attacks Review Steganography

07

NYU

Affine Cipher

- Extension of the shift cipher: rather than just adding the key to the plaintext, we also multiply by the key
- We use for this a key consisting of two parts: k = (a, b)

Let k, x, y $\in \{0,1, \ldots, 25\}$

- Encryption: $y = e_k(x) \equiv a\ x + b\ mod\ 26$
- Decryption: $x = d_k(x) \equiv a^{-1}(y - b)\ mod\ 26$

NYU

Affine Cipher

Suppose $a \in Z_m$

$a^{-1} \bmod m$ (the multiplicative inverse of a modulo m):

$aa^{-1} \equiv a^{-1}a \equiv 1 \pmod m$

$M = C = Z_{26}$

$K=\{(a,b) \in Z_{26} X Z_{26}\}$

For $k=(a,b) \in K$; $x, y \in Z_{26}$

$e_K(x)=(ax+b) \bmod 26$

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

NYU

Affine Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
13	14	15	16	17	18	19	20	21	22	23	24	25

NYU

An example of Affine Cipher

- For example, let a = 9 and b = 2, so we are working with 9x + 2. Take a plaintext letter such as h (= 7). It is encrypted to 9 . 7 + 2 = 65 = 13 (mod 26), which is the letter N . Using the same function, we obtain

  Affine → CVVWPM

NYU

Affine Cipher: How do we decrypt?

- If we were working with rational numbers rather than mod 26, we would start with $y = 9x + 2$ and solve: $x = 1/9 (y - 2)$.
- But 1/9 needs to be reinterpreted when we work mod 26.
- Since gcd (9,26) = 1, there is a multiplicative inverse for 9 (mod 26).
- In fact, $9 \cdot 3 = 1 \pmod{26}$, so 3 is the desired inverse and can be used in place of 1/9. We therefore have

$$x = 3 (y - 2) = 3y - 6 = 3y + 20 \pmod{26}.$$

- The letter V (= 21) is mapped to $3 \cdot 21+20 = 83 = 5 \pmod{26}$, which is the letter f.
- Similarly, we see that the ciphertext CVVWPM is decrypted back to affine.

Affine Cipher: one-to-one decrypt

Suppose we try to use the function 13x + 4 as our encryption function.
We obtain

input → ?

If we alter the input, we obtain

alter → ?

- Clearly this function leads to errors. It is impossible to decrypt, since several plaintexts yield the same ciphertext.
- In particular, we note that encryption must be one-to-one, and this fails in the present case.

NYU

Affine Cipher: one-to-one decrypt

What goes wrong in this example?

- If we solve $y = 13x + 4$, we obtain $x = 1/13(y - 4)$. But $1/13$ does not exist mod 26 since $\gcd(13,26) = 13 \neq 1$.
- More generally, it can be shown that $a \cdot x + b$ is a one-to-one function mod 26 if and only if $\gcd(a, 26) = 1$.
- In this case, decryption uses $x = a^{-1} \cdot y - a^{-1} \cdot b \pmod{26}$, where $a \cdot a^{-1} = 1 \pmod{26}$.
- So decryption is also accomplished by an affine function.

NYU

Affine Cipher

Since the inverse of a is needed for inversion, we can only use values for a for which:

$$\gcd(a, 26) = 1$$

There are 12 values for a that fulfill this condition. From this follows that the key space is only 12 x 26 = 312

Again, several attacks are possible, including:

- Exhaustive key search and letter frequency analysis, similar to the attack against the substitution cipher

Affine Cipher: possible attacks

Ciphertext only:

- An exhaustive search through all 312 keys would take longer than the corresponding search in the case of the shift cipher; however, it would be very easy to do on a computer.
- When all possibilities for the key are tried, a fairly short ciphertext, say around 20 characters, will probably correspond to only one meaningful plaintext, thus allowing the determination of the key. It would also be possible to use frequency counts, though this would require much longer texts.

Affine Cipher: possible attacks

Chosen plaintext:

- Choose ab as the plaintext.
  - The first character of the ciphertext will be a . 0 + b = b,
  - and the second will be a + b.

Therefore, we can find the key.

Chosen ciphertext:

- Choose AB as the ciphertext.
- This yields the decryption function of the form $x = a_1y + b_1$.
- We could solve for y and obtain the encryption key.

NYU

Affine Cipher: possible attacks

Known plaintext:
With a little luck, knowing two letters of the plaintext and the corresponding letters of the ciphertext suffices to find the key.
In any case, the number of possibilities for the key is greatly reduced and a few more letters should yield the key.

For example, suppose the plaintext starts with *if* and the corresponding ciphertext is PQ.
In numbers, this means that 8 (= i) maps to 15 (= P) and 5 maps to 16. Therefore, we have the equations $8a + b = 15$ and $5a + b = 16 \pmod{26}$.
Subtracting yields $3a = -1 = 25 \pmod{26}$, which has the unique solution $a = 17$.
Using the first equation, we find $8 \cdot 17 + b = 15 \pmod{26}$, which yields $b = 9$.

NYU

Affine Cipher: possible attacks

Known plaintext:

Suppose instead that the plaintext *go* corresponds to the ciphertext TH. We obtain the equations 6a + b = 19 and 14a + b = 7 (mod 26). Subtracting yields - 8a = 12 (mod 26). Since gcd (- 8 ,26) = 2, this has two solutions: a = 5,18.

The corresponding values of b are both 15 (this is not a coincidence; it will always happen this way when the coefficients of a in the equations are even).

So we have two candidates for the key: (5,15) and (18,15). However, gcd(18, 26) ≠1 so the second is ruled out. Therefore, the key is (5,15).

NYU

Affine Cipher: possible attacks

Known plaintext:

If we know only one letter of plaintext, we still get a relation between a and b. For example, if we only know that g in plaintext corresponds to T in ciphertext, then we have $6a + b = 19 \pmod{26}$. There are 12 possibilities for a and each gives one corresponding b. Therefore, an exhaustive search through the 12 keys should yield the correct key.

Cryptography vs Steganography

- Cryptography
  - Scrambling information so it cannot be read
  - Transforms information into secure form so unauthorized persons cannot access it
- Steganography
  - Hides the existence of data
  - An image, audio, or video file can contain hidden messages embedded in the file
  - Achieved by dividing data and hiding in unused portions of the file

Definitions

- Cryptography
  - Scrambling information so it cannot be read
  - Transforms information into secure form so unauthorized persons cannot access it
- Steganography
  - Hides the existence of data
  - An image, audio, or video file can contain hidden messages embedded in the file
  - Achieved by dividing data and hiding in unused portions of the file

Fundamental Properties

A fundamental tradeoff exists between three key variables: robustness, capacity and imperceptibility which restricts steganography designers.

- Imperceptibility is the perceptual similarity between the host and stego. In audio steganography, imperceptibility is evaluated as an audible distortion caused by signal modifications.
- The ability of embedded data or watermark for withstanding against intentional and unintentional attacks is measured as robustness.
- Capacity (Payload) indicates the amount of data that can successfully be embedded without introducing perceptual distortion.

Substitution-based Steganography

Substitution-based algorithms replace insignificant bits of the original file with message data in a manner that the least amount of distortion is caused.

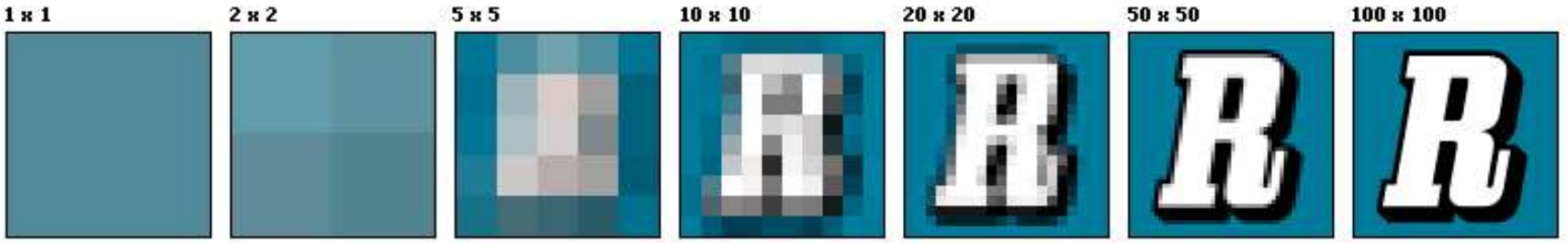
Why Substitution Technique? Mostly the payload of substitution techniques is more than 40,000 bps, while more robust techniques, like spread spectrum, has a negligible payload that is only about 4 bps.

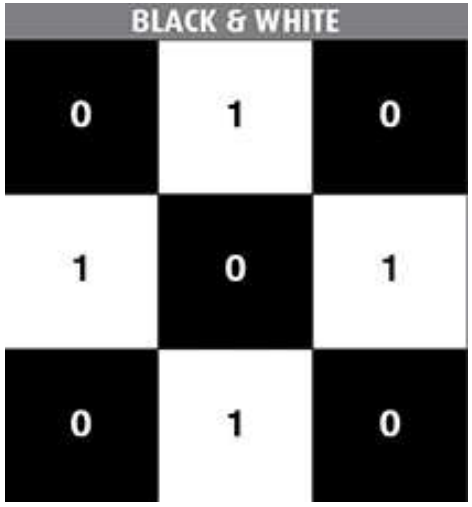
- Audio Least Significant Bit (LSB) steganography takes advantage of the quantization error that usually derives from the task of digitizing the audio signal.

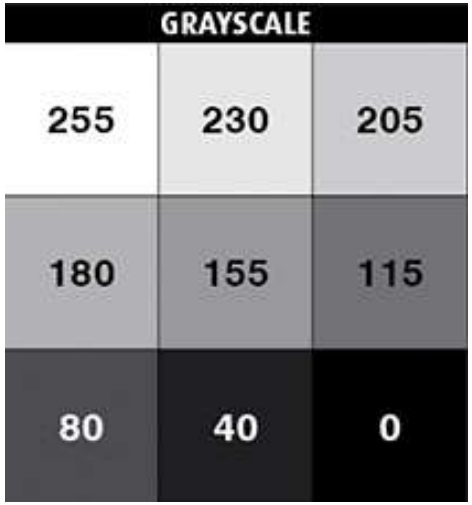
- The information is encoded into the right most bits per samples or least significant bits from audio data.

NYU

Pixels and Resolution

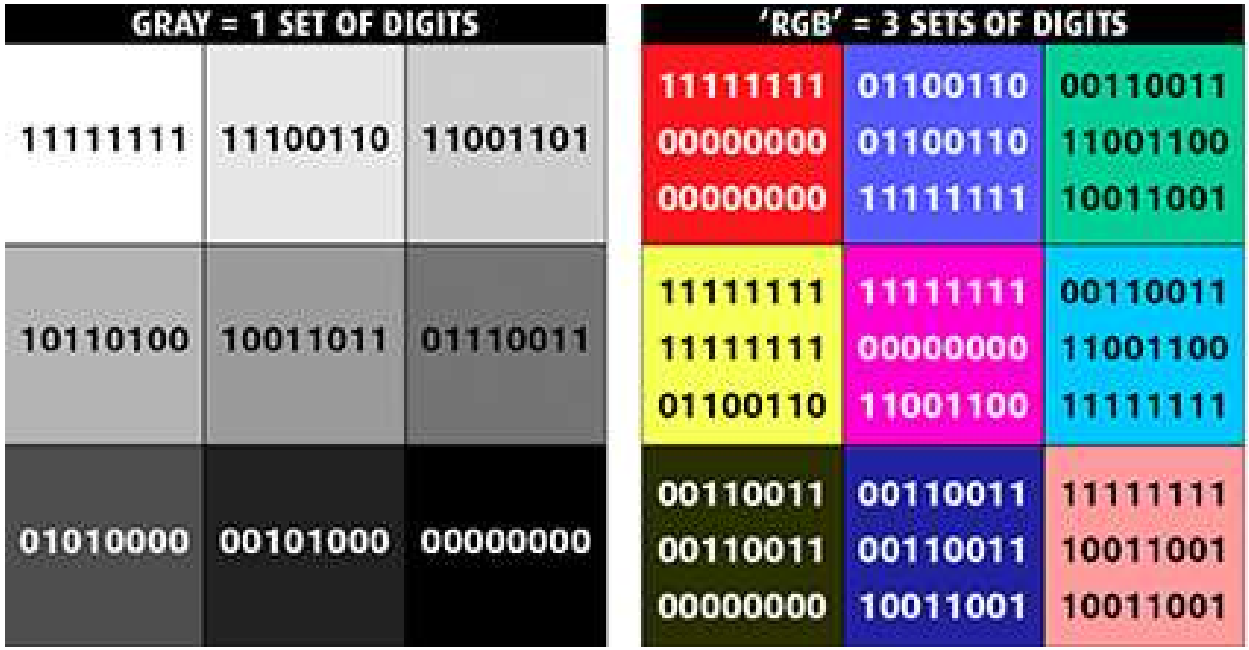




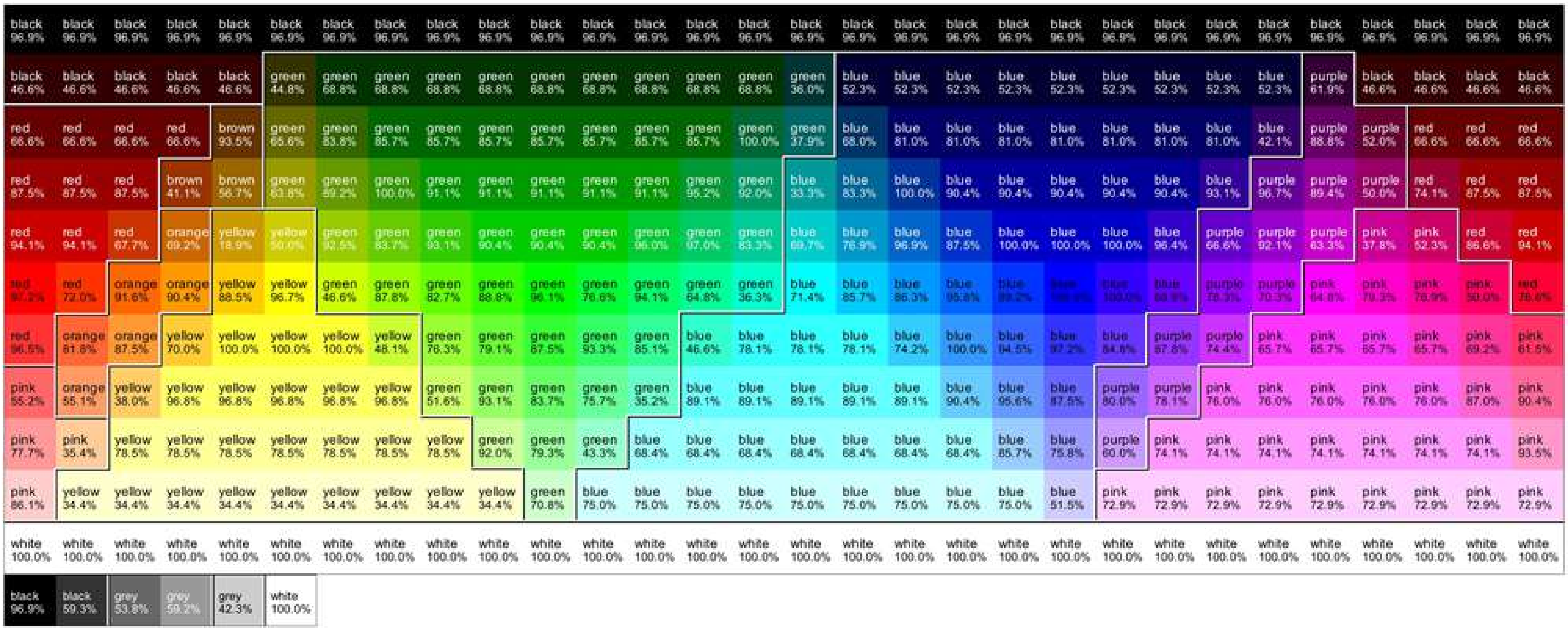


NYU

Set of digits



Colour spectrum



Steganography - Least Significant Bits (LSB) Technique

- Suppose you have a 1024*768 pixel picture with 32bit RGB system.
- If you specify the intensity of blue colour by 7 bits instead, how much will be the capacity of this picture to hide a watermark?

NYU

Trade-off

It is not achievable to get a high payload and high-quality technique at the same time in steganography. A trade-off between the payload and the quality (imperceptibility and robustness) is necessary.

Apart from robustness that principally is not desirable for substitution techniques; the only remaining measure to achieve a high payload system, is imperceptibility.

However, substitution techniques comparatively are well-known in achieving high capacity, but the distortion caused by substitution degrades the quality.

Therefore, to take advantage of a potential high payload, imperceptibility should be retained.

Three Bit per Sample Correction

In each sample, three LSB bits are used for the message bits embedding. So that, any alteration in the sample must not change those three bits.

✓ If either of following possibilities occur:

- Host bits are 101 and message bits are 000
- Host bits are 110 and message bits are 000 or 001
- Host bits are 111 and message bits are 000 or 001 or 010

Following algorithm below will modify the sample toward decreasing the amount of error.

From (fourth bit in the byte) TO (the end of the OR a bit whose value is 1)

DO reset current bit to 1;

IF (the loop above got finished because of second condition)

DO reset current bit to 0;

NYU

Three Bit per Sample Correction

- ✓ All possible modification for the following possibility were studied, and found out that there is no chance to modify.
  - Host bits are 011 OR host bits are 100
- ✓ If either of following possibilities occur:
  - Host bits are 010 and message bits are 111
  - Host bits are 001 and message bits are 110 or 111
  - Host bits are 000 and message bits are 101 or 110 or 111

Following algorithm below will modify the sample toward decreasing the amount of error.

From (fourth bit in the byte) TO (the end of byte OR a bit whose value is 0)

DO reset current bit to 0;

IF (the loop above got finished because of second condition)

DO reset current bit to 1;

NYU

Definitions

Original host chromosome: $10000000_2=128_{10}$

Embedding message bits: 1111 (4 bps payload is supposed)

Initial stego chromosome: $10001111_2=143_{10}$

Current difference: 143-128= 15

1st iteration of mutation procedure: $10011111_2=159_{10}$

Current difference: 159-128= 31 (getting worse)

2nd iteration of mutation procedure: $10111111_2=191_{10}$

Current difference: 191-128= 63 (getting worse)

3rd iteration of mutation procedure: $11111111_2=255_{10}$

Current difference: 255-128= 127 (getting worse)

4th iteration of mutation procedure: $01111111_2=127_{10}$

Current difference: 128-127= 1 (The best!)

NYU

Demo