Geometry based Encryption Algorithm

Cryptology – Ceaser to AES

Symmetric Encryption - Same Key

Asymmetric Encryption – Public and Private keys

Stream Cipher – One Byte at a time (Ceaser, RC4, FISH)

Block Cipher – 8bits to 512Bytes at a time (AES, RC5, Blowfish)

To keep in mind

- Attacks
- Key Length
- Authentication
- Performance

Attacks

Linear Cryptanalysis – Linear Algebra

Differential Cryptanalysis – Non random behaviour of algorithm

Related-Key attack – Relationship between the keys

Weak Keys – Undesired behaviour of algorithm

Known Plain-Text attack – knows the Plain and Cipher text

Cipher Text-only attack – some knowledge about the plain text

Key Length

Key Size	Possible combinations
1-bit	2
2-bit	4
4-bit	16
8-bit	256
16-bit	65536
32-bit	4.2×10^{9}
56-bit (DES)	7.2×10^{16}
64-bit	1.8×10^{19}
128-bit (AES)	3.4×10^{38}
192-bit (AES)	6.2×10^{57}
256-bit (AES)	1.1×10^{77}

Key size	Time to Crack
56-bit	399 seconds
128-bit	1.02×10^{18} years
192-bit	1.872×10^{37} years
256-bit	3.31×10^{56} years

Brute Force Attack

Exponential increase in possible combinations as the key size increases

Difference between cracking the 128 algorithm and 256 algorithm is considered minimal

Authentication

Why Authenticate?

- Noise and Surety

Ways to Authenticate:

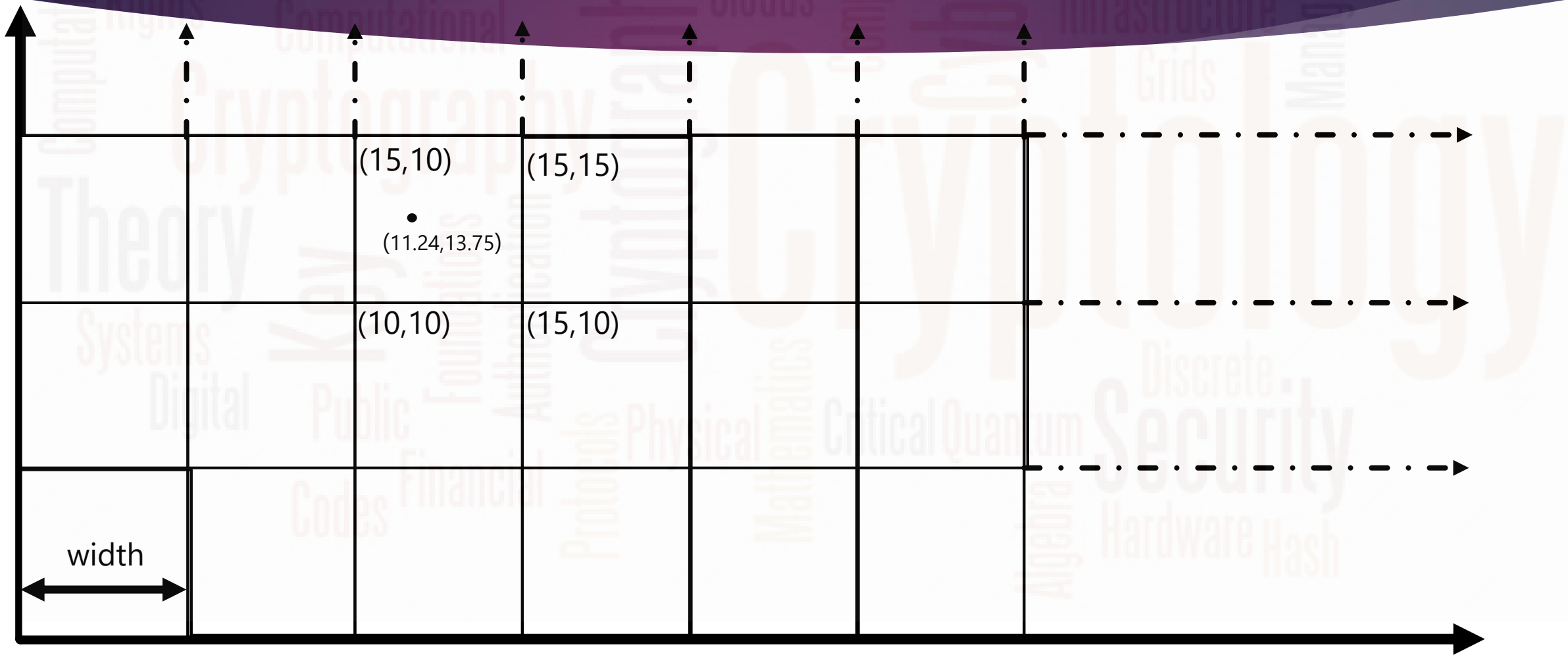
- MAC then Encrypt
- Encrypt and MAC
- Encrypt then MAC

WHAT IS NEW?

- Mapping characters/Blocks to points in X-Y plane
- Symmetric Algorithm
- Multiple Keys
- No more pseudo random numbers
- Keys length up to 128bits
- Can encrypt file of any type (No need to convert images to strings)
- A Block Cipher that can be very easily parallelized

How does it work?

Codebook



Other configuration option

- Width = 5 units
- FPPB (Floating Point Precision Bits) = 8
- Total characters = 2 ** 8 = 256
- Xmax = 500
- Ymax = 500

ASSUMPTION

- Both the sender and receiver have same Codebook.
- Both the sender and receiver know all the configuration options.

Encryption

- Assume we want to encrypt a block of size 1byte (8 bits)
- Let the byte value be 65
- We choose any 2 random numbers less than xmax and ymax respectively.
- Let us call them intX and intY with values 11 and 13 respectively.
- We choose another 2 random numbers keeping in mind our precision values.
- We will call them floatX and floatY with values 024 and 079 respectively.
- Find the key from the Codebook using intX and intY.

Encryption

- Key = Codebook[10][10] = (1001 1100 0101 1010)b = 40026
- Concatenate floatX and floatY to create a 16bit number say floatXY
  - floatXY = 0001 1000 0100 1011 = 6219
  - floatXYXor = floatXY ^ key = 33809
- Find Beta = floatXYXor % (2**8) = 17
- Find diff = Byte – Beta = 65-17 = 48

Encryption

- newFloatXYXor = floatXYXor + diff = 33809+48 = 33857
- newFloatXY = newFloatXYXor ^ key = 6171
- Separate [newFloatX,newFloatY] = [newfloatXY] = (0001 1000 0001 1011)b
- newFloatX = 00011000 = 24, newFloatY = 0001 1000 0001 1011 = 27
- Point encrypted is 11.024 , 13.027

Decryption

- Point (x,y) = (11.24,13.27)
- intX = 12 , intY = 13, floatX = 024, floatY = 027
- Key = Codebook[10][10] = 1001 1100 0101 1010 = 40026
- floatXY = 0001 1000 0001 1011 = 6171
- floatXYXor = floatXY ^ key = 33857
- Byte = floatXYXor % totalCharacters = 33857 % 256 = 65
- Byte = 65

Block Sizes

- We just saw the implementation of Block size of 1byte/8bits
- The same has been extended to Block size of 8bytes/64bits
- Bigger the block size higher the performance
- The key length = 2 * Block size so a key of 128bits has been successfully tested.
- As instead of working on characters we are working on bytes we can encrypt/decrypt any type of file.
- Successfully checked on txt,docx,png,mp3,mkv and bin file formats.

Performance

We are planning to run the other algorithms code on our system to get an actual difference.

Current Performance :
(64Bit block size)

→ 256 MB in 55 seconds
→ 4.65MB/s
→ 4880.65 KB/s

Text File Size In Kbytes	DES	Triple-DES	Blowfish	Propose Algorithm	AES
20	20	34	25	23	42
48	30	55	27	26	55
100	47	81	33	30	90
247	83	111	45	43	112
321	90	167	46	44	164
694	144	226	47	45	210
899	240	230	64	56	256
910	245	299	68	63	213
Average Time	112.375	150.00	44.375	41.25	142.75
Throughput (Kilobyte/sec)	4.16	3.11	11.16	11.33	3.27

(i5-intel, Dual Core, 2.2GHz, 8GB)

(AMD Athlon TF-20, 1.6GHz, 3GB)

Future Work

- Increase Key/Block Size
- Code Optimisation
- Padding
- Performance Comparison
- Avalanche Effect
- Authentication
- Paper