CS6500 Assignment 1

J. Prem Krishnaa

August 2017

1 OpenSSL

The objective of this assignment is to understand and use the OpenSSL API for invoking existing cryptographic algorithmic implementations. In this assignment we measure the performance metrics namely, mean block encryption time and mean block decryption time by varying algorithm, key size, mode and file sizes. We also give the time required for a successful brute force attack based on the extrapolation of the average times obtained. Apart from that we use SHA256 algorithm to generate key from the user pass phrase by taking first 'n' bits of the hash as per the command line key size. We also initialize the IV for 64 bit block (DES, 3DES CBC) by taking last 8 bytes of the obtained hash and similarly by taking last 16 bytes of the hash for 128 bit block (AES CBC).

2 System Configuration

The following system configuration was obtained using the command `lscpu`:

- Architecture: x86_64, CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 4, On-line CPU(s) list: 0-3
- Thread(s) per core: 2, Core(s) per socket: 2
- Socket(s): 1, NUMA node(s): 1
- Vendor ID: GenuineIntel, CPU family: 6, Model: 69
- Model name: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz
- Memory: 3.8 GiB, Stepping: 1
- CPU MHz: 951.093, BogoMIPS: 4788.61
- Virtualization: VT-x
- L1d cache: 32K, L1i cache: 32K
- L2 cache: 256K, L3 cache: 3072K
- NUMA node0 CPU(s): 0-3

3 Results

Total time, Mean time/block are in microseconds. Time taken for brute force attack is in years.

Operation	File Size	Mode	Key Size	No of blocks	Total Time	Mean time/block	Bruteforce
Enc	10KB	CBC	128 bits	625	56	0.089	$9.6x10^{23}$
Dec	10KB	CBC	128 bits	625	23	0.037	$3.99x10^{23}$
Enc	100KB	CBC	128 bits	6250	203	0.033	$3.56x10^{23}$
Dec	100KB	CBC	128 bits	6250	70	0.011	$1.18x10^{23}$
Enc	10KB	CBC	192 bits	625	34	0.054	$1.07x10^{43}$
Dec	10KB	CBC	192 bits	625	17	0.027	$5.37x10^{42}$
Enc	100KB	CBC	192 bits	6250	235	0.038	$7.56x10^{42}$
Dec	100KB	CBC	192 bits	6250	69	0.011	$2.18x10^{42}$
Enc	10KB	ECB	128 bits	625	15	0.024	$2.58x10^{23}$
Dec	10KB	ECB	128 bits	625	16	0.026	$2.80x10^{23}$
Enc	100KB	ECB	128 bits	6250	64	0.010	$1.07x10^{23}$
Dec	100KB	ECB	128 bits	6250	61	0.009	$9.71x10^{22}$
Enc	10KB	ECB	192 bits	625	16	0.026	$5.17x10^{42}$
Dec	10KB	ECB	192 bits	625	17	0.027	$5.37x10^{42}$
Enc	100KB	ECB	192 bits	6250	64	0.010	$1.99x10^{42}$
Dec	100KB	ECB	192 bits	6250	70	0.011	$2.18x10^{42}$
Enc	10KB	CTR	128 bits	625	18	0.029	$3.12x10^{23}$
Dec	10KB	CTR	128 bits	625	14	0.022	$2.37x10^{23}$
Enc	100KB	CTR	128 bits	6250	62	0.010	$1.07x10^{23}$
Dec	100KB	CTR	128 bits	6250	60	0.009	$9.71x10^{22}$
Enc	10KB	CTR	192 bits	625	18	0.029	$5.77x10^{42}$
Dec	10KB	CTR	192 bits	625	16	0.026	$5.17x10^{42}$
Enc	100KB	CTR	192 bits	6250	70	0.011	$2.18x10^{42}$
Dec	100KB	CTR	192 bits	6250	64	0.010	$1.99x10^{42}$

Table 1: Performance Metrics for AES

Operation	File Size	Mode	Key Size	No of blocks	Total Time	Mean time/block	Bruteforce
Enc	10KB	CBC	56 bits	1250	180	0.144	329
Dec	10KB	CBC	56 bits	1250	171	0.137	313
Enc	100KB	CBC	56 bits	12500	1711	0.137	313
Dec	100KB	CBC	56 bits	12500	1632	0.131	299
Enc	10KB	ECB	56 bits	1250	176	0.141	322
Dec	10KB	ECB	56 bits	1250	200	0.160	365
Enc	100KB	ECB	56 bits	12500	1665	0.133	303
Dec	100KB	ECB	56 bits	12500	1749	0.140	319

Table 2: Performance Metrics for DES

Operation	File Size	Mode	Key Size	No of blocks	Total Time	Mean time/block	Bruteforce
Enc	10KB	CBC	112 bits	1250	466	0.373	$6.14x10^{19}$
Dec	10KB	CBC	112 bits	1250	453	0.362	$5.96x10^{19}$
Enc	100KB	CBC	112 bits	12500	5210	0.417	$6.86x10^{19}$
Dec	100KB	CBC	112 bits	12500	4497	0.359	$5.91x10^{19}$
Enc	10KB	CBC	168 bits	1250	567	0.454	$5.38x10^{36}$
Dec	10KB	CBC	168 bits	1250	456	0.357	$4.23x10^{36}$
Enc	100KB	CBC	168 bits	12500	4883	0.391	$4.63x10^{36}$
Dec	100KB	CBC	168 bits	12500	4491	0.359	$4.26x10^{36}$
Enc	10KB	ECB	112 bits	1250	465	0.372	$6.12x10^{19}$
Dec	10KB	ECB	112 bits	1250	470	0.376	$6.19x10^{19}$
Enc	100KB	ECB	112 bits	12500	4818	0.385	$6.33x10^{19}$
Dec	100KB	ECB	112 bits	12500	4616	0.369	$6.07x10^{19}$
Enc	10KB	ECB	168 bits	1250	469	0.375	$4.45x10^{36}$
Dec	10KB	ECB	168 bits	1250	469	0.375	$4.45x10^{36}$
Enc	100KB	ECB	168 bits	12500	4598	0.359	$4.26x10^{36}$
Dec	100KB	ECB	168 bits	12500	5151	0.412	$4.88x10^{36}$

Table 3: Performance Metrics for 3DES

Using known plain text attack, we can do brute force attack by both encryption and decryption operations. The time taken for brute force attack is given by 2^n x Mean time/block for each encryption/decryption, where n is the key size in bits.

4 Inferences

Let us first talk about performance metrics for AES. In CBC mode for a given key size, file size, mean block encryption time is larger (2x) than mean block decryption time (This is because in CBC mode decryption to decrypt a block of cipher text we don't need the plaintext decrypted from the previous block unlike the case during encryption. Hence decryption can be parallelized). We also see that for a given file size, there is not much difference to the metrics on changing key size since the block size remains same and the number of rounds doesn't differ much either (10 rounds for 128 bit key, 12 rounds for 192 bit key). On increasing file size, we observe that the metrics decrease (probably because of pre-computed encryptions being stored in a hash table somewhere). In ECB mode, we observe that the mean block operation time for encryption and decryption are almost same for a given file size. Similar to CBC, we observe that the metrics decrease as we increase the file size which can be again attributed to pre-computed encryptions. Also the encryption metrics for ECB are smaller than that of CBC as expected since ECB can be parallelized while CBC cannot be. In CTR mode we observe comparable metrics to that of ECB mode, since CTR mode can also be parallelized but also takes advantage from CBC mode in being more secure. The observation on key size made for CBC mode applies to ECB and CTR modes as well.

Now looking at DES performance metrics, we can observe that the mean block encryption/decryption times are much larger than that of AES. This is because of larger block size of AES and level of optimization in the software code, architecture used and given that AES is the current standard. We can see that in CBC mode decryption is slightly faster as observed in AES CBC mode. In ECB mode, encryption and decryption times are almost same as expected with encryption slightly faster than that of CBC mode.

Coming to 3DES, we observe that the metrics are almost thrice that obtained in DES which is again expected attributing to the 3 DES modules. Rest of the inferences made for AES and DES apply here as well. In case of 112 bit key, we have $K_1 = K_3$, hence it could be made slightly faster by storing the computed encryptions/decryptions in a table. Apart from all these, we observe that when we increase the key size, we are more secure from brute force attacks as expected.