

CPE 449: ENCRYPTION ASSIGNMENT

DAN OTIENO

09/01/2023.

Task 1:

No submission required.

Task 2:

Task 2: Encryption mode - ECB vs. CBC

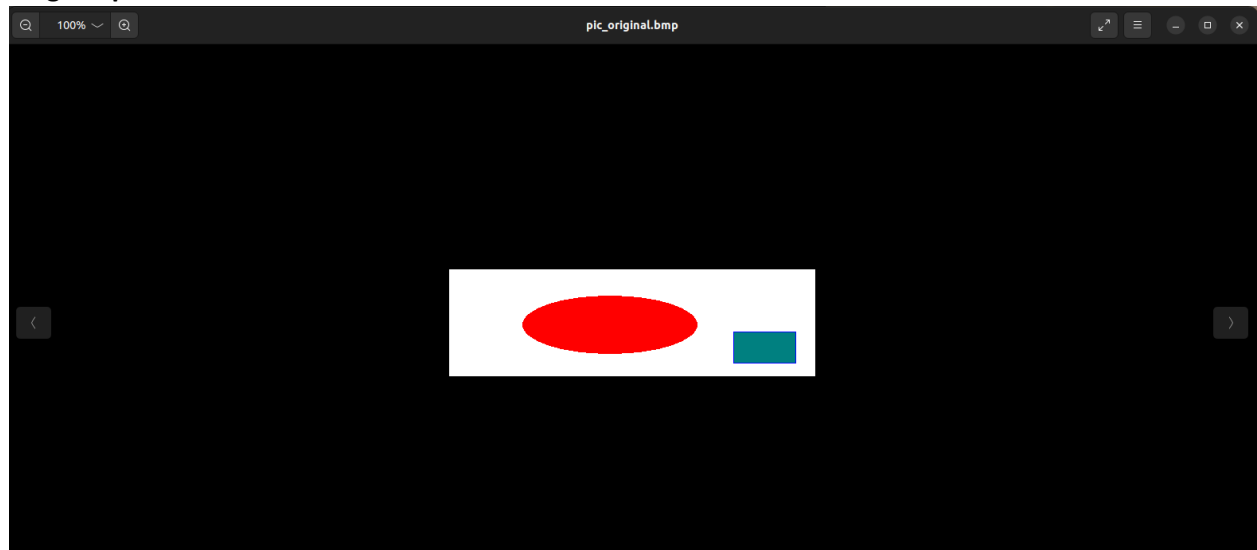
The file `pic_original.bmp` [pic_original.bmp](#) contains a simple picture. Encrypt this picture, so people without the encryption keys cannot know what is in the picture. Please encrypt the file using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes, and then do the following:

- Treat the encrypted picture as a picture and use a picture viewing software to display it. For a .bmp file, the first 54 bytes contain the header information about the picture. Add the plain text header bytes to the encrypted file so it will be displayed as a legitimate .bmp file. Replace the header of the encrypted picture with that of the original picture. You can use a hex editor tool (e.g. ghex or Bless) to directly modify binary files.
- Display the encrypted picture using any picture viewing software.

Submit:

1. Submit the ECB encrypted photo (insert the photo in your lab report document) after replacing the header so that it is viewable in an image viewer.
2. Can you derive any useful information about the original picture from the encrypted picture? Please explain your observations.

1. Original photo:



Encrypted photo (aes-128-ecb):



Encrypted photo (aes-128-cbc):



2. There's not much information that can be derived when it comes to the colors from the original image. However, whether anything can be determined from the original image depends on the encryption method. When the ECB method is used, we can see the exact shapes from the original image, whereas, in the second encryption method, CBC, the original image is fully encrypted such that there is no visible data from the original. Because the CBC method carries over encryption information from the one block and uses it to encrypt the next block, whereas ECB encrypts each block independently, CBC is considered a more secure method between the two, and that is demonstrated in the exercise above.

Task 3:

Task 3: Encryption Mode – Corrupted Cipher Text

To understand the properties of various encryption modes, we would like to do the following exercise:

- Create a text file that is at least 64 bytes long.
- Encrypt the file using the AES-128 cipher.
- Change exactly 1 bit in the encrypted output using a hex editor.
- Decrypt the corrupted file (encrypted version) using the correct key and IV.

Submit: Please answer the following questions:

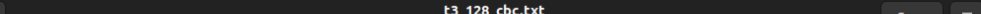
1. What information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively?
2. Please explain why for each algorithm in the previous step.
3. What are the implications of these differences?

Original text:

```
t3_original.txt
~/Desktop/CPE/CPE_449/Hw_1/Task_3

1 AA BB CC DD EE FF GG HH
2 II JJ KK LL MM NN OO PP
3 QQ RR SS TT UU V|
```

Aes-128 Encryptions (CBC, CFB, ECB, OFB respectively):

[illegible][illegible]

Aes-128 Decryptions after modifying 1 bit per file (CBC, CFB, ECB, OFB respectively):

```
t3_128_cbc_dec.txt
~/Desktop/CPE/CPE_449/Hw_1/Task_3

1 11 22 33
2 5.^ 6.##;_iiúF GG(HH
3 II JJ KK LL MM NN OO PP
4 QQ RR SS TT UU V|
```

```
t3_128_cfb_dec.txt
~/Desktop/CPE/CPE_149/Hw_1/Task_3
1 AA MB CC DD EE F9 2 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031
```

Open Save

t3_128_ecb_dec.txt
~/Desktop/CPE/CPE_449/Hw_1/Task_3

```

1 00000000000000000000000000000000 GG HH
2 II JJ KK LL MM NN OO PP
3 QQ RR SS TT UU V

```

```
Open  t3_128_ofb_dec.txt  Save  - /Desktop/CPE/CPE_449/Hw_1/Task_3
1 AA CC DD EE FF GG HH
2 II JJ KK LL MM NN OO PP
3 QQ RR SS TT UU V
```

1. All the cypher files are modified in some way when decrypted. We get varying amounts of data recovered from the corrupted cypher files. 48 bytes of data is recovered from the CBC cypher, 46 bytes recovered from CFB cypher, 47 bytes recovered from the ECB cypher, and finally, 63 bytes recovered from the OFB cypher.
2. The most data recovered from the corrupted cypher is from the OFB-encrypted file. This is because the algorithm uses a stream of bits to encrypt subsequent data blocks, the decryption is completed one bit at a time, such that only the text corresponding to the corrupted bit is affected. The CBC and CFB modes are similar in that during decrypting of a ciphertext block, one should add XOR the output data received from the decryption algorithm to the previous ciphertext block. Because the receiver knows all the ciphertext blocks just after obtaining the encrypted message, he can decrypt the message using many threads simultaneously. So, if one bit is corrupted, then all the corresponding bits are affected when the file is decrypted. For the ECB method, each block of text/ciphertext is encrypted or decrypted separately, so all corresponding bits are affected if one is corrupted as well. (Source: <https://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html>)
3. Because of the differences in how much of the original data can be retrieved from a corrupted cypher text file, the implications depend on the circumstances, in the case where bad actors are involved in decrypting the file, the method of encryption/decryption is important because less data can be recovered to prevent information ending up in the

wrong hands. However, in the case where extremely important encrypted information ends up with the rightful receiver, the loss of data can lead to misinterpretation or misrepresentation of the original message, leading to detrimental outcomes.

Task 4:

Task 4 : Padding

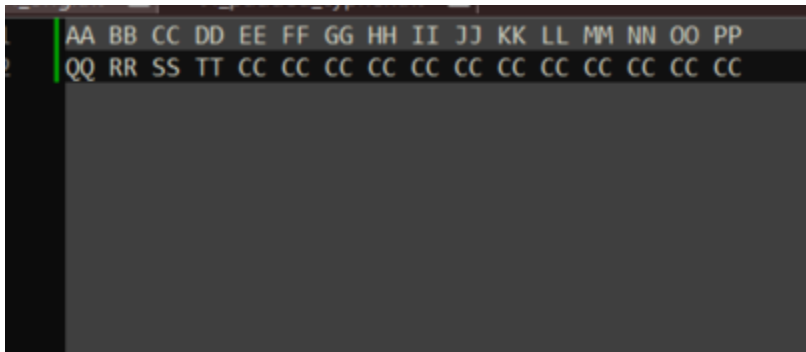
PKCS#7 pads input text at the trailing end with $k - (l \bmod k)$ octets all having value $k - (l \bmod k)$, where l is the length of the input and k is the block length.

Design a scheme to alter cipher text so that when decryption occurs the padding is left behind (hint: you may increase the length of the cipher text).

Submit:

1. Describe your scheme.
2. Show the padding for a 20-byte plain text file. Insert of a screen shot showing the padding from a hex editor.
3. How many bytes of padding are added for a 32-byte file.
4. Show the padding for a 32-byte plain text file. Insert of a screen shot showing the padding from a hex editor.

Original text (with padding):



Hex view of original text:

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	41	41	20	42	42	20	43	43	20	44	44	20	45	45	20	46	AA BB CC DD EE F
00000010	46	20	47	47	20	48	48	20	49	49	20	4a	4a	20	4b	4b	F GG HH II JJ KK
00000020	20	4c	4c	20	4d	4d	20	4e	4e	20	4f	4f	20	50	50	20	LL MM NN OO PP
00000030	0d	0a	51	51	20	52	52	20	53	53	20	54	54	20	43	43	..QQ RR SS TT CC
00000040	20	43	43	20	43	43	20	43	43	20	43	43	20	43	43	20	CC CC CC CC CC
00000050	43	43	20	43	43	20	43	43	20	43	43	20	43	43	20	43	CC CC CC CC CC C
00000060	43																C

Padded cypher:



Hex view of padded cypher:

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	db	01	0e	96	8d	0e	67	0c	20	41	6a	02	4c	84	58	ea	Û...-..g. Aj.L,,Xê
00000010	56	bc	42	cf	46	74	83	4e	98	8f	37	b5	d3	b5	d1	36	V¼BİFt fN..7µÓµÑ6
00000020	4d	a0	1c	ee	16	78	86	d5	3d	67	e3	2c	c8	02	e1	41	M..İ.xİÖ=gã,Ê.áA
00000030	ce	89	b3	f7	a5	7f	ac	ad	2b	3d	72	a6	7b	36	2e	f6	Î%³÷¥.İ-+=r!{6.ö
00000040	45	0f	17	e3	d1	c8	a1	24	c3	27	be	20	f4	b4	dc	c9	E..añE; \$Ã'¾ ö'ÜÉ
00000050	ba	b4	b4	c9	06	d3	4c	83	6d	50	e4	74	38	0f	f3	6d	°´´É.ÓLfmPät8.óm
00000060	82	fa	0e	d4	65	5c	df	db	da	d3	d3	fe	f4	86	a4	82	,ú.Ôe\8ÛÛÓÓöpö†µ,

Hex view of modified bytes in padded cypher:

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	02	e1	41áA
00000030	ce	89	b3	f7	a5	7f	ac	ad	2b	3d	72	a6	7b	36	2e	f6	Î%³÷¥.İ-+=r!{6.ö
00000040	45	0f	17	e3	d1	c8	a1	24	c3	27	be	20	f4	b4	dc	c9	E..añE; \$Ã'¾ ö'ÜÉ
00000050	ba	b4	b4	c9	06	d3	4c	83	6d	50	e4	74	38	0f	f3	6d	°´´É.ÓLfmPät8.óm
00000060	82	fa	0e	d4	65	5c	df	db	da	d3	d3	fe	f4	86	a4	82	,ú.Ôe\8ÛÛÓÓöpö†µ,

Decrypted cypher file after altering bytes such that only the padding is viewable:

