

```
1
2 # *Version: FTPChat 1.0
3
4 # *This product is from 'AOSP'
5
6 from datetime import datetime
7 from ftplib import FTP
8
9 CHARACTERS = (
10     "a",
11     "b",
12     "c",
13     "d",
14     "e",
15     "f",
16     "g",
17     "h",
18     "i",
19     "j",
20     "k",
21     "l",
22     "m",
23     "n",
24     "o",
25     "p",
26     "q",
27     "r",
28     "s",
29     "t",
30     "u",
31     "v",
32     "w",
33     "x",
34     "y",
35     "z",
36     "A",
37     "B",
38     "C",
39     "D",
40     "E",
41     "F",
42     "G",
43     "H",
44     "I",
45     "J",
46     "K",
47     "L",
48     "M",
49     "N",
50     "O",
51     "P",
52     "Q",
53     "R",
54     "S",
55     "T",
56     "U",
57     "V",
58     "W",
59     "X",
60     "Y",
61     "Z",
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      "0" /
95      "1" /
96      "2" /
97      "3" /
98      "4" /
99      "5" /
100     "6" /
101     "7" /
102     "8" /
103     "9" /
104 )
105 KEY_1 = (
106     "E" /
107     "|" /
108     "9" /
109     "~" /
110     ")" /
111     "s" /
112     "C" /
113     "1" /
114     "/" /
115     "!" /
116     "o" /
117     "<" /
118     "$" /
119     "]" /
120     "M" /
121     "z" /
122     "X" /
123     "\\\" /
124     "Z" /
125     "5" /
126     "Q" /
127     "S" /
128     "h" /
129     "t" /
130     "e" /
131     "7" /
132     "}" /
133     "?" /
134     "O" /
```

```
135      "i" /
136      "Y" /
137      "j" /
138      "G" /
139      "-" /
140      "v" /
141      "V" /
142      "%" /
143      "P" /
144      "R" /
145      "[" /
146      "p" /
147      "r" /
148      "g" /
149      "U" /
150      "@" /
151      "(" /
152      "=" /
153      "W" /
154      "A" /
155      ">" /
156      "f" /
157      "B" /
158      "d" /
159      "H" /
160      "q" /
161      "'" /
162      "#" /
163      "{" /
164      "u" /
165      "I" /
166      "+" /
167      "3" /
168      "4" /
169      "`" /
170      "N" /
171      "L" /
172      ";" /
173      "J" /
174      "8" /
175      "m" /
176      "w" /
177      "6" /
178      "k" /
179      "-" /
180      "." /
181      "K" /
182      "b" /
183      "D" /
184      "&" /
185      "0" /
186      "x" /
187      "," /
188      ":" /
189      "c" /
190      "T" /
191      "^" /
192      "2" /
193      "1" /
194      "F" /
195      "Y" /
196      "*" /
197      "a" /
198      "n" /
199
200 )
201 KEY_2 = (
```

202       ", "/  
203       "u" /  
204       "i" /  
205       "5" /  
206       ">" /  
207       "6" /  
208       "3" /  
209       "7" /  
210       "L" /  
211       "<" /  
212       "=" /  
213       "H" /  
214       "]" /  
215       "´" /  
216       "E" /  
217       "R" /  
218       "8" /  
219       "\\\" /  
220       "4" /  
221       "B" /  
222       "m" /  
223       "0" /  
224       "!" /  
225       "Z" /  
226       "|" /  
227       "^" /  
228       "Y" /  
229       "X" /  
230       "O" /  
231       "T" /  
232       "f" /  
233       "M" /  
234       "c" /  
235       "W" /  
236       "N" /  
237       "G" /  
238       ": " /  
239       "p" /  
240       "w" /  
241       "1" /  
242       "Q" /  
243       "(" /  
244       "q" /  
245       "U" /  
246       "S" /  
247       "C" /  
248       "." /  
249       "d" /  
250       "&" /  
251       "\_" /  
252       "2" /  
253       "s" /  
254       "9" /  
255       "g" /  
256       ")" /  
257       "k" /  
258       "-" /  
259       "A" /  
260       "K" /  
261       "[" /  
262       "e" /  
263       "/" /  
264       "\*" /  
265       "?" /  
266       "I" /  
267       "~" /  
268       "l" /

```
269 "z" /
270 "$" /
271 "j" /
272 "#" /
273 "v" /
274 "D" /
275 "@" /
276 "h" /
277 "i" /
278 "+" /
279 "b" /
280 "J" /
281 "F" /
282 "r" /
283 "V" /
284 "{" /
285 "}" /
286 ";" /
287 "a" /
288 "t" /
289 "P" /
290 "o" /
291 "%" /
292 "x" /
293 "I" /
294 "Y" /
295 "n" /
296 )
297 KEY_3 = (
298 "6" /
299 "i" /
300 "W" /
301 "=" /
302 "F" /
303 "$" /
304 "T" /
305 "8" /
306 "C" /
307 "+" /
308 "!" /
309 "l" /
310 "G" /
311 "<" /
312 "a" /
313 "(" /
314 "J" /
315 "b" /
316 "K" /
317 "2" /
318 "Z" /
319 "7" /
320 "Q" /
321 "M" /
322 "f" /
323 "-" /
324 "o" /
325 "P" /
326 "5" /
327 "j" /
328 "r" /
329 "@" /
330 "m" /
331 "_" /
332 "p" /
333 ";" /
334 "X" /
335 "x" /
```

```
336      "." /
337      "R" /
338      "[" /
339      "A" /
340      ":" /
341      "S" /
342      "?" /
343      "O" /
344      "V" /
345      "U" /
346      "|" /
347      "w" /
348      "Y" /
349      ">" /
350      "c" /
351      "N" /
352      "z" /
353      "e" /
354      "D" /
355      "}" /
356      "*" /
357      ")" /
358      "v" /
359      "s" /
360      "H" /
361      "`" /
362      "%" /
363      "L" /
364      "/" /
365      "y" /
366      "3" /
367      "q" /
368      "I" /
369      "n" /
370      "9" /
371      "^" /
372      "#" /
373      "d" /
374      "]" /
375      "\\\" /
376      "0" /
377      "i" /
378      "g" /
379      "t" /
380      "1" /
381      "u" /
382      "E" /
383      "B" /
384      "4" /
385      "i" /
386      "k" /
387      "&" /
388      "h" /
389      "/" /
390      "~" /
391      "{" /
392  )
393  KEY_4 = (
394      "1" /
395      "." /
396      "&" /
397      "q" /
398      "Q" /
399      "l" /
400      "m" /
401      "Z" /
402      "Y" /
```

403 ":" /  
404 "!" /  
405 "C" /  
406 "|" /  
407 "i" /  
408 "A" /  
409 "r" /  
410 "O" /  
411 "}" /  
412 "5" /  
413 "x" /  
414 "J" /  
415 ">" /  
416 "d" /  
417 "K" /  
418 "!" /  
419 ")" /  
420 "]" /  
421 "F" /  
422 "V" /  
423 "j" /  
424 "@" /  
425 "t" /  
426 "4" /  
427 "E" /  
428 "[" /  
429 "p" /  
430 "7" /  
431 "\$" /  
432 "h" /  
433 "c" /  
434 "M" /  
435 "6" /  
436 "/" /  
437 "!" /  
438 "," /  
439 "`" /  
440 "9" /  
441 "\\\" /  
442 "3" /  
443 "s" /  
444 "D" /  
445 "S" /  
446 "<" /  
447 "=" /  
448 "b" /  
449 ";" /  
450 "\*" /  
451 "k" /  
452 "2" /  
453 "N" /  
454 "L" /  
455 "I" /  
456 "e" /  
457 "H" /  
458 "X" /  
459 "8" /  
460 "{" /  
461 "w" /  
462 "z" /  
463 "W" /  
464 "P" /  
465 "U" /  
466 "0" /  
467 "?" /  
468 "%" /  
469 "v" /

```
470     "g" ,
471     "n" ,
472     "R" ,
473     "o" ,
474     "-" ,
475     "T" ,
476     "_" ,
477     "G" ,
478     "^" ,
479     "~" ,
480     "a" ,
481     "+" ,
482     "u" ,
483     "y" ,
484     "(" ,
485     "B" ,
486     "#" ,
487     "f" ,
488 )
489 KEY_5 = (
490     "O" ,
491     "p" ,
492     "/" ,
493     "1" ,
494     "X" ,
495     "3" ,
496     ">" ,
497     "5" ,
498     "Q" ,
499     "n" ,
500     "T" ,
501     "}" ,
502     "@" ,
503     "!" ,
504     "#" ,
505     "&" ,
506     "P" ,
507     "=" ,
508     "g" ,
509     "b" ,
510     "d" ,
511     "C" ,
512     "f" ,
513     "U" ,
514     "A" ,
515     "D" ,
516     "%" ,
517     "q" ,
518     "G" ,
519     "K" ,
520     "\\\" ,
521     "`" ,
522     "!" ,
523     "|" ,
524     "k" ,
525     "S" ,
526     "R" ,
527     "(" ,
528     "y" ,
529     "6" ,
530     "^" ,
531     "Y" ,
532     ")" ,
533     "*" ,
534     "j" ,
535     "z" ,
536     "l" ,
```



```
537 "J" /
538 "N" /
539 "C" /
540 "W" /
541 "-" /
542 "e" /
543 "M" /
544 "2" /
545 "<" /
546 "h" /
547 "]" /
548 "?" /
549 ":" /
550 "!" /
551 "7" /
552 "9" /
553 "H" /
554 "F" /
555 "w" /
556 "L" /
557 "Z" /
558 "s" /
559 "$" /
560 "x" /
561 "B" /
562 "0" /
563 "[" /
564 "I" /
565 "{" /
566 "m" /
567 "-" /
568 "r" /
569 "E" /
570 "." /
571 "t" /
572 "~" /
573 "a" /
574 "8" /
575 "i" /
576 "+" /
577 "u" /
578 "v" /
579 "o" /
580 "," /
581 "V" /
582 "4" /
583 ";" /
584 )
585 KEY_6 = (
586 ">" /
587 "S" /
588 "<" /
589 "B" /
590 "O" /
591 "[" /
592 "Y" /
593 "7" /
594 "0" /
595 "?" /
596 "!" /
597 "*" /
598 "X" /
599 "W" /
600 "i" /
601 "D" /
602 "8" /
603 "6" /
```

604 "4" /  
605 "i" /  
606 "r" /  
607 "j" /  
608 "f" /  
609 "+" /  
610 "c" /  
611 "v" /  
612 "P" /  
613 " " /  
614 " $\overline{V}$ " /  
615 "H" /  
616 "d" /  
617 "q" /  
618 "T" /  
619 "R" /  
620 "w" /  
621 "h" /  
622 "M" /  
623 "Z" /  
624 "J" /  
625 "Y" /  
626 "." /  
627 "," /  
628 "K" /  
629 "u" /  
630 "g" /  
631 "&" /  
632 "`" /  
633 "#" /  
634 "s" /  
635 "(" /  
636 ":" /  
637 "e" /  
638 "p" /  
639 "\\\" /  
640 "^" /  
641 "U" /  
642 "n" /  
643 "C" /  
644 "}" /  
645 "G" /  
646 "-" /  
647 "g" /  
648 "z" /  
649 "Q" /  
650 "=" /  
651 "N" /  
652 "\$" /  
653 ";" /  
654 "2" /  
655 "]" /  
656 "|" /  
657 "E" /  
658 "~" /  
659 "k" /  
660 "L" /  
661 "!" /  
662 "3" /  
663 "1" /  
664 "b" /  
665 "/" /  
666 "x" /  
667 "t" /  
668 "5" /  
669 "l" /  
670 ")" /

```
671      "%",
672      "F",
673      "A",
674      "O",
675      "a",
676      "{",
677      "@",
678      "m",
679      "I",
680  )
681  KEY_7 = (
682      "Y",
683      ":",
684      "q",
685      "S",
686      "O",
687      "+",
688      "W",
689      "w",
690      "|",
691      "m",
692      "e",
693      "%",
694      "F",
695      ";",
696      "\\ ",
697      "R",
698      "J",
699      "I",
700      "a",
701      "i",
702      "G",
703      "/",
704      "U",
705      "8",
706      "1",
707      "x",
708      "~",
709      "B",
710      "v",
711      "#",
712      "Z",
713      "?",
714      "A",
715      "^",
716      "@",
717      "X",
718      ",",
719      "7",
720      "o",
721      "s",
722      "' ",
723      ".",
724      "<",
725      "5",
726      "C",
727      "[",
728      "*",
729      "$",
730      "&",
731      "g",
732      "j",
733      ">",
734      "4",
735      "b",
736      "-",
737      "H",
```

```
738 " _ " /
739 " r " /
740 " z " /
741 " E " /
742 " 6 " /
743 " T " /
744 " f " /
745 " d " /
746 " 9 " /
747 " ( " /
748 " { " /
749 " h " /
750 " D " /
751 " ) " /
752 " Q " /
753 " O " /
754 " ` " /
755 " N " /
756 " 2 " /
757 " p " /
758 " l " /
759 " v " /
760 " K " /
761 " k " /
762 " t " /
763 " c " /
764 " ] " /
765 " } " /
766 " = " /
767 " ' " /
768 " n " /
769 " L " /
770 " Y " /
771 " M " /
772 " P " /
773 " 3 " /
774 " ! " /
775 " u " /
776 )
777 KEY_8 = (
778 " > " /
779 " ' " /
780 " w " /
781 " + " /
782 " ) " /
783 " & " /
784 " y " /
785 " D " /
786 " ( " /
787 " M " /
788 " Z " /
789 " . " /
790 " v " /
791 " r " /
792 " * " /
793 " T " /
794 " [ " /
795 " s " /
796 " e " /
797 " n " /
798 " K " /
799 " 0 " /
800 " C " /
801 " 5 " /
802 " f " /
803 " L " /
804 " / " /
```

805 "I" /  
806 "l" /  
807 "`" /  
808 "?" /  
809 "W" /  
810 "m" /  
811 "x" /  
812 "p" /  
813 "3" /  
814 "H" /  
815 ", " /  
816 "\$" /  
817 "A" /  
818 "B" /  
819 "<" /  
820 "S" /  
821 "X" /  
822 "{" /  
823 "]" /  
824 "!" /  
825 ":" /  
826 "4" /  
827 "R" /  
828 "6" /  
829 "O" /  
830 "a" /  
831 "J" /  
832 "|" /  
833 "q" /  
834 "@" /  
835 "7" /  
836 "}" /  
837 "g" /  
838 " " /  
839 "⋀" /  
840 "%" /  
841 "N" /  
842 "\\\" /  
843 "Y" /  
844 "1" /  
845 "k" /  
846 "E" /  
847 "i" /  
848 "c" /  
849 "-" /  
850 "P" /  
851 "o" /  
852 "Q" /  
853 "~" /  
854 "F" /  
855 "2" /  
856 "#" /  
857 "z" /  
858 "G" /  
859 "h" /  
860 ";" /  
861 "9" /  
862 "d" /  
863 "U" /  
864 "u" /  
865 "t" /  
866 "j" /  
867 "i" /  
868 "8" /  
869 "b" /  
870 "v" /  
871 "=" /

```
872 )
873 KEY_9 = (
874     "g" ,
875     "9" ,
876     "z" ,
877     "n" ,
878     "w" ,
879     "p" ,
880     "[" ,
881     "=" ,
882     "J" ,
883     "a" ,
884     "}" ,
885     "t" ,
886     "Q" ,
887     "Y" ,
888     "V" ,
889     " " ,
890     "G" ,
891     "^" ,
892     "2" ,
893     "{" ,
894     "k" ,
895     "u" ,
896     "S" ,
897     ";" ,
898     "0" ,
899     "L" ,
900     "A" ,
901     "-" ,
902     "8" ,
903     "1" ,
904     "%" ,
905     "|" ,
906     "\\\" ,
907     "i" ,
908     "!" ,
909     "U" ,
910     "f" ,
911     "+" ,
912     "D" ,
913     "(" ,
914     "<" ,
915     "O" ,
916     "r" ,
917     "C" ,
918     "P" ,
919     "o" ,
920     "R" ,
921     "H" ,
922     "c" ,
923     "#" ,
924     "6" ,
925     "4" ,
926     "x" ,
927     "!" ,
928     "h" ,
929     ":" ,
930     "3" ,
931     "5" ,
932     "," ,
933     "!" ,
934     "&" ,
935     "N" ,
936     "j" ,
937     "$" ,
938     "@" ,
```

```
939      "I" /
940      "F" /
941      "m" /
942      "q" /
943      ") " /
944      "B" /
945      ". " /
946      "s" /
947      "e" /
948      "j " /
949      "E" /
950      "d" /
951      "X" /
952      "Z" /
953      "/" /
954      "M" /
955      "K" /
956      "W" /
957      "?" /
958      "T" /
959      "Y" /
960      "l " /
961      ">" /
962      "v" /
963      "7" /
964      "` " /
965      "b" /
966      "*" /
967      "~" /
968  )
969
970  KEY_10 = (
971      "n" /
972      "f" /
973      ">" /
974      "p" /
975      "3" /
976      "F" /
977      "\\ " /
978      ":" /
979      "D" /
980      "h" /
981      "` " /
982      ") " /
983      " ( " /
984      "<" /
985      "O" /
986      "R" /
987      "P" /
988      "u" /
989      "y" /
990      "8" /
991      "I" /
992      "!" /
993      "2" /
994      "*" /
995      "t" /
996      "v" /
997      "Y" /
998      "&" /
999      "/" /
1000     "S" /
1001     "L" /
1002     "+" /
1003     "5" /
1004     ". " /
1005     "k" /
```

```
1006     "m",
1007     "G",
1008     "{ ",
1009     "Z",
1010     "W",
1011     "O",
1012     "A",
1013     "e",
1014     "; ",
1015     "E",
1016     "U",
1017     "q",
1018     "[ ",
1019     "N",
1020     "s",
1021     "^",
1022     "x",
1023     "C",
1024     "=",
1025     "| ",
1026     "g",
1027     "J",
1028     "c",
1029     "~",
1030     "b",
1031     ", ",
1032     "$",
1033     "T",
1034     "]" ",
1035     "9",
1036     "Q",
1037     "a",
1038     "r",
1039     "4",
1040     "H",
1041     "v",
1042     "@",
1043     "-",
1044     "j",
1045     "i",
1046     "X",
1047     "6",
1048     "7",
1049     "}",
1050     "B",
1051     "?",
1052     "#",
1053     "M",
1054     "z",
1055     "!",
1056     "d",
1057     "W",
1058     "!",
1059     " ",
1060     "K",
1061     "o",
1062     "%",
1063     "l",
1064     "1",
1065 )
1066
1067
1068 # *Encryption layers func.
1069
1070
1071 def reverser(text):
1072     return text[::-1]
```



```
1073
1074
1075 def mac1_encode(text):
1076     result = []
1077     for char in text:
1078         if char in CHARACTERS:
1079             index = CHARACTERS.index(char)
1080             result.append(KEY_1[index])
1081         else:
1082             result.append(char)
1083     return "".join(result)
1084
1085
1086 def mac1_decode(text):
1087     result = []
1088     for char in text:
1089         if char in KEY_1:
1090             index = KEY_1.index(char)
1091             result.append(CHARACTERS[index])
1092         else:
1093             result.append(char)
1094     return "".join(result)
1095
1096
1097 def mac2_encode(text):
1098     result = []
1099     for char in text:
1100         if char in CHARACTERS:
1101             index = CHARACTERS.index(char)
1102             result.append(KEY_2[index])
1103         else:
1104             result.append(char)
1105     return "".join(result)
1106
1107
1108 def mac2_decode(text):
1109     result = []
1110     for char in text:
1111         if char in KEY_2:
1112             index = KEY_2.index(char)
1113             result.append(CHARACTERS[index])
1114         else:
1115             result.append(char)
1116     return "".join(result)
1117
1118
1119 def mac3_encode(text):
1120     result = []
1121     for char in text:
1122         if char in CHARACTERS:
1123             index = CHARACTERS.index(char)
1124             result.append(KEY_3[index])
1125         else:
1126             result.append(char)
1127     return "".join(result)
1128
1129
1130 def mac3_decode(text):
1131     result = []
1132     for char in text:
1133         if char in KEY_3:
1134             index = KEY_3.index(char)
1135             result.append(CHARACTERS[index])
1136         else:
1137             result.append(char)
1138     return "".join(result)
1139
```

```
1140
1141 def mac4_encode(text):
1142     result = []
1143     for char in text:
1144         if char in CHARACTERS:
1145             index = CHARACTERS.index(char)
1146             result.append(KEY_4[index])
1147         else:
1148             result.append(char)
1149     return "".join(result)
1150
1151
1152 def mac4_decode(text):
1153     result = []
1154     for char in text:
1155         if char in KEY_4:
1156             index = KEY_4.index(char)
1157             result.append(CHARACTERS[index])
1158         else:
1159             result.append(char)
1160     return "".join(result)
1161
1162
1163 def mac5_encode(text):
1164     result = []
1165     for char in text:
1166         if char in CHARACTERS:
1167             index = CHARACTERS.index(char)
1168             result.append(KEY_5[index])
1169         else:
1170             result.append(char)
1171     return "".join(result)
1172
1173
1174 def mac5_decode(text):
1175     result = []
1176     for char in text:
1177         if char in KEY_5:
1178             index = KEY_5.index(char)
1179             result.append(CHARACTERS[index])
1180         else:
1181             result.append(char)
1182     return "".join(result)
1183
1184
1185 def mac6_encode(text):
1186     result = []
1187     for char in text:
1188         if char in CHARACTERS:
1189             index = CHARACTERS.index(char)
1190             result.append(KEY_6[index])
1191         else:
1192             result.append(char)
1193     return "".join(result)
1194
1195
1196 def mac6_decode(text):
1197     result = []
1198     for char in text:
1199         if char in KEY_6:
1200             index = KEY_6.index(char)
1201             result.append(CHARACTERS[index])
1202         else:
1203             result.append(char)
1204     return "".join(result)
1205
1206
```

```
1207 def mac7_encode(text):
1208     result = []
1209     for char in text:
1210         if char in CHARACTERS:
1211             index = CHARACTERS.index(char)
1212             result.append(KEY_7[index])
1213         else:
1214             result.append(char)
1215     return "".join(result)
1216
1217
1218 def mac7_decode(text):
1219     result = []
1220     for char in text:
1221         if char in KEY_7:
1222             index = KEY_7.index(char)
1223             result.append(CHARACTERS[index])
1224         else:
1225             result.append(char)
1226     return "".join(result)
1227
1228
1229 def mac8_encode(text):
1230     result = []
1231     for char in text:
1232         if char in CHARACTERS:
1233             index = CHARACTERS.index(char)
1234             result.append(KEY_8[index])
1235         else:
1236             result.append(char)
1237     return "".join(result)
1238
1239
1240 def mac8_decode(text):
1241     result = []
1242     for char in text:
1243         if char in KEY_8:
1244             index = KEY_8.index(char)
1245             result.append(CHARACTERS[index])
1246         else:
1247             result.append(char)
1248     return "".join(result)
1249
1250
1251 def mac9_encode(text):
1252     result = []
1253     for char in text:
1254         if char in CHARACTERS:
1255             index = CHARACTERS.index(char)
1256             result.append(KEY_9[index])
1257         else:
1258             result.append(char)
1259     return "".join(result)
1260
1261
1262 def mac9_decode(text):
1263     result = []
1264     for char in text:
1265         if char in KEY_9:
1266             index = KEY_9.index(char)
1267             result.append(CHARACTERS[index])
1268         else:
1269             result.append(char)
1270     return "".join(result)
1271
1272
1273 def mac10_encode(text):
```

```

1274     result = []
1275     for char in text:
1276         if char in CHARACTERS:
1277             index = CHARACTERS.index(char)
1278             result.append(KEY_10[index])
1279         else:
1280             result.append(char)
1281     return "".join(result)
1282
1283
1284 def mac10_decode(text):
1285     result = []
1286     for char in text:
1287         if char in KEY_10:
1288             index = KEY_10.index(char)
1289             result.append(CHARACTERS[index])
1290         else:
1291             result.append(char)
1292     return "".join(result)
1293
1294
1295 def first_main_encryption(text):
1296     layer1 = mac5_encode(text)
1297     layer2 = reverser(layer1)
1298     layer3 = mac4_encode(layer2)
1299     layer4 = mac2_encode(layer3)
1300     layer5 = mac1_encode(layer4)
1301     layer6 = mac3_encode(layer5)
1302     layer7 = mac8_encode(layer6)
1303     layer8 = mac7_encode(layer7)
1304     layer9 = mac9_encode(layer8)
1305     layer10 = mac6_encode(layer9)
1306     layer11 = mac8_encode(layer10)
1307     return str(mac10_encode(layer11))
1308
1309
1310 def first_main_decryption(text):
1311     layer1 = mac10_decode(text)
1312     layer2 = mac8_decode(layer1)
1313     layer3 = mac6_decode(layer2)
1314     layer4 = mac9_decode(layer3)
1315     layer5 = mac7_decode(layer4)
1316     layer6 = mac8_decode(layer5)
1317     layer7 = mac3_decode(layer6)
1318     layer8 = mac1_decode(layer7)
1319     layer9 = mac2_decode(layer8)
1320     layer10 = mac4_decode(layer9)
1321     layer11 = reverser(layer10)
1322     return str(mac5_decode(layer11))
1323
1324
1325 def encrypt(text):
1326     return str(
1327         first_main_encryption(first_main_encryption(text)))
1328
1329
1330 def decrypt(text):
1331     return str(
1332         first_main_decryption(first_main_decryption(text)))
1333
1334
1335 def send_message(username, message, ftp_host, ftp_user, ftp_pass, chat_input, chat_name):
1336     """Send a message to the chat file on the FTP server."""
1337     try:
1338         if message == "REFRESH":
1339             msg = encrypt(f"{username}: REFRESH")
1340         else:

```

```

1341     msg = encrypt(f"{datetime.now()}:{username}:{message}\n")
1342     ftp = FTP(ftp_host)
1343     ftp.login(ftp_user, ftp_pass)
1344     try:
1345         with open(chat_input, "wb") as file:
1346             ftp.retrbinary(f"RETR {chat_name}", file.write)
1347     except:
1348         pass
1349
1350     with open(chat_input, "a", encoding="utf-8") as file:
1351         file.write(f"{msg}\n")
1352
1353     with open(chat_input, "rb") as file:
1354         ftp.storbinary(f"STOR {chat_name}", file)
1355
1356     ftp.quit()
1357 except Exception as error:
1358     print(f"Error sending message: {error}")
1359
1360
1361 def read_messages(ftp_host, ftp_user, ftp_pass, chat_input, chat_name):
1362     """Read messages from the chat file on the FTP server."""
1363     try:
1364         with FTP(ftp_host) as ftp:
1365             ftp.login(ftp_user, ftp_pass)
1366             with open(chat_input, "wb") as file:
1367                 ftp.retrbinary(f"RETR {chat_name}", file.write)
1368             with open(chat_input, "r", encoding="utf-8") as file:
1369                 lines = file.readlines()
1370                 if not lines:
1371                     print("No messages yet.")
1372                     return
1373                 print("-----Chat-----")
1374                 for line in lines:
1375                     decrypted_message = decrypt(line.strip())
1376                     print(decrypted_message)
1377                 print("-----")
1378     except Exception as error:
1379         print(f"Error reading messages. ({error})")
1380
1381
1382 def multiline_input(prompt):
1383     """Get multiline input from the user until 'END' is entered."""
1384     print(prompt)
1385     lines = []
1386     while True:
1387         line = input()
1388         if line.upper() == "END":
1389             break
1390         lines.append(line)
1391     return "\n".join(lines)
1392
1393
1394 print(r"""
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
```

```
1408         break
1409
1410     ftp_user = input("Please, type the FTP username:\n")
1411     if not ftp_user:
1412         print("No FTP User entered, Exiting the app")
1413         break
1414
1415     ftp_pass = input("Please, type the FTP password:\n")
1416     if not ftp_pass:
1417         print("No Password entered, Exiting the app")
1418         break
1419
1420     username = input("Please, type your username:\n")
1421     if not username:
1422         username = "Anonymous"
1423         print("Username is empty, using 'Anonymous' as default.")
1424         continue
1425
1426     chat_input = input("Please, type the chat file name:\n") + ".txt"
1427     if not chat_input:
1428         print("No chat file name input, exiting the program")
1429         break
1430
1431     chat_name = f"/usb1_1/{chat_input}"
1432     all_filled = True
1433     break
1434
1435
1436 if all_filled:
1437     print("""
1438 Important Notes before starting:
1439 1. Type 'END' to finish your message.
1440 2. Type 'REFRESH' to refresh messages.
1441 3. Don't turn off your FTP Server while using this chat.
1442 """)
1443
1444 while True:
1445     read_messages(ftp_host, ftp_user, ftp_pass, chat_input, chat_name)
1446     message = multiline_input(
1447         "Type your message:\n")
1448     send_message(username, message, ftp_host, ftp_user,
1449                 ftp_pass, chat_input, chat_name)
1450
```