Задача A. Oracle Padding Attack. Easy.

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сегодня Вы взломаете современный шифр AES с CBC, и причем минимальными и реалистичными средствами. Представим, что Вы работаете в штабе, и для спасения Родины важно узнать, находится ли в перехваченном шифротексте сообщение о начале атаки ("`Yes`"), отказе от атаки ("`No`"), или отсутствии решения на данный момент ("`N/A`"). Дополнительная помощь в расшифровке минимальна, удалось получить узнать лишь адрес интерфейса к одному из центральных серверов врага, и ему можно прислать шифротекст, а в ответ придет или сообщение об успешном приеме ("`Ok`"), или сообщение об ошибке. В данном случае, единственной ошибкой, которая может нам помочь, является неправильное заполнение блока. Но даже такого оракула, впрочем, достаточно для взлома шифра.

Согласно стандарту $PKCS\#5$, если блок размером 16 байт шифрует сообщение размером 15 байт, то его надо дополнить одним байтом $0x01$, в случае сообщения размером 14 байт, дополнением будет два байта $0x02, 0x02$. Если блок шифрует сообщение размером в блок (16 байт), то сообщение дополняется целым блоком, которое содержит 16 байт, $0x10, \ldots, 0x10$.

В нашем случае сообщение помещаются в 1 блок, и в качестве шифротекста фигурируют два блока $C_0 = IV, C_1$. Расшифровка оракулом осуществляется как $m||p := IV \oplus Dec(C_1)$, где m - сообщение, а p - его дополнение; после расшифровки делается проверка p.

Протокол взаимодействия

Интерактор выбирает сообщение, которое он будет кодировать, из трёх вариантов: "`Yes`", "`No`" и "`N/A`". Затем случайным образом генерирует AES ключ и IV в 16 байт, и кодирует строку AES шифрованием с CBC и PKCS5 заполнением. Интерактор пишет в двух строках закодированное сообщение и IV в Base64.

Затем происходит не более 1000 раундов, за которые нужно взломать шифр. В каждом раунде взаимодействия происходит 3 шага:

1. Вы выводите "`YES`", если можете расшифровать сообщение, и "`NO`", иначе.
2. Если Вы расшифровали сообщение, следующая строка должна его содержать. ("`Yes`", "`No`" или "`N/A`").
3. В противном случае, Вы можете сделать запрос. В первой строке должно содержаться сообщение на расшифровку в Base64, а во второй строке IV, с которым будет расшифровка, также в Base64.
4. В ответ вы получите или "`Ok`" или "`Wrong padding`" в зависимости от успешности декодирования.

Пример

стандартный ввод	стандартный вывод
r9MKX1L3KIus9lNxv9Dyxg== 9sktozrwHU+3cOmMAyXOHQ== Wrong padding Ok ... 	 NO r9MKX1L3KIus9lNxv9Dyxg== zv2buxkdiC5cTRuuYjeKAA== NO r9MKX1L3KIus9lNxv9Dyxg== zv2buxkdiC5cTRuuYjeKEQ== ... YES Yes

Задача B. Oracle Padding Attack. Hard.

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Сегодня Вы взломаете современный шифр AES с CBC, и причем минимальными и реалистичными средствами. Представим, что Вы работаете в штабе, и для спасения Родины важно узнать, находится ли в перехваченном шифротексте сообщение о начале атаки ("`Yes`"), отказе от атаки ("`No`"), или отсутствии решения на данный момент ("`N/A`"). Дополнительная помощь в расшифровке минимальна, удалось получить узнать лишь адрес интерфейса к одному из центральных серверов врага, и ему можно прислать шифротекст, а в ответ придет или сообщение об успешном приеме ("`Ok`"), или сообщение об ошибке. В данном случае, единственной ошибкой, которая может нам помочь, является неправильное заполнение блока. Но даже такого оракула, впрочем, достаточно для взлома шифра.

Согласно стандарту $PKCS\#5$, если блок размером 16 байт шифрует сообщение размером 15 байт, то его надо дополнить одним байтом $0x01$, в случае сообщения размером 14 байт, дополнением будет два байта $0x02, 0x02$. Если блок шифрует сообщение размером в блок (16 байт), то сообщение дополняется целым блоком, которое содержит 16 байт, $0x10, \ldots, 0x10$.

В нашем случае сообщение помещаются в 1 блок, и в качестве шифротекста фигурируют два блока $C_0 = IV, C_1$. Расшифровка оракулом осуществляется как $m||p := IV \oplus Dec(C_1)$, где m - сообщение, а p - его дополнение; после расшифровки делается проверка p.

Протокол взаимодействия

Интерактор выбирает сообщение, которое он будет кодировать, из трёх вариантов: "`Yes`", "`No`" и "`N/A`". Затем случайным образом генерирует AES ключ и IV в 16 байт, и кодирует строку AES шифрованием с CBC и PKCS5 заполнением. Интерактор пишет в двух строках закодированное сообщение и IV в Base64.

Затем происходит три раунда, за которые нужно взломать шифр. В каждом раунде взаимодействия происходит 3 шага:

1. Вы выводите "`YES`", если можете расшифровать сообщение, и "`NO`", иначе.
2. Если Вы расшифровали сообщение, следующая строка должна его содержать. ("`Yes`", "`No`" или "`N/A`").
3. В противном случае, Вы можете сделать запрос. В первой строке должно содержаться сообщение на расшифровку в Base64, а во второй строке IV, с которым будет расшифровка, также в Base64.
4. В ответ вы получите или "`Ok`" или "`Wrong padding`" в зависимости от успешности декодирования.

Пример

стандартный ввод	стандартный вывод
r9MKX1L3KIus9lNxv9Dyxg== 9sktozrwHU+3cOmMAyXOHQ== Wrong padding Ok ... 	 NO r9MKX1L3KIus9lNxv9Dyxg== zv2buxkdiC5cTRuuYjeKAA== NO r9MKX1L3KIus9lNxv9Dyxg== zv2buxkdiC5cTRuuYjeKEQ== ... YES Yes