Кодер паролей для их защиты при хранении в записных книжках - passcoder.

H1

Требуется спроектировать и реализовать систему защиты паролей при их хранении в записных книжках или блокнотах.

Требования к программе.

H2

1. Должны поддерживаться три режима работы: режим регистрации пользователя, режим шифрования пароля, режим расшифрования пароля;
2. В режиме регистрации пользователя программа должны генерировать открытый и секретный ключи для шифрования паролей криптосистемой Рабина. Открытый ключ должен быть подписан ЭЦП RSA на секретном ключе программы.
3. Предусмотреть процедуру смены ключа подписи программы.
4. Предусмотреть защиту секретного ключа пользователя паролем.
5. В режиме шифрования пароля запускается интерактивный сеанс, в котором программа запрашивает путь к файлу с открытым ключом и просит ввести пароль. В результате программа выводит пароль, зашифрованный на открытом ключе пользователя криптосистемой Рабина.
6. Шифр-текст пароля выводится в кодировке base32, в которой удобно хранить его в записной книге или в блокноте.
7. В режиме расшифрования пароля программа запрашивает путь к файлу с секретным ключом и шифр-текст пароля в кодировке base32. На выходе программа выводит на экран пароль. В случае, если секретный ключ не соответствует шифр-тексту, то необходимо вывести сообщение об этом пользователю.
8. (ОПЦИЯ). Реализовать хранение зашифрованных паролей в текстовом файле следующего формата. Каждая запись — с новой строки. Запись содержит три поля: («Порядковый номер записи», «Описание пароля», «шифр-текст пароля в кодировке base32»). Требуется реализовать возможность отобразить пользователю файл паролей и выбор пользователем пароля для расшифрования с помощью указания порядкового номера пароля.
9. Предпочтительный язык программирования — python. В случае использования другого языка нужно создать простую сборочную системы на основе make-файлов для трёх операционных системы: Windows 10, MacOS 10.15+, Linux.

Требования к выполнению задания и принцип его оценивания.

H2

1. Все технические решения нужно обосновать с точки зрения надёжности и стойкости к известным автору атакам (вклад в оценку — 45%). Предполагается, что к программе будет приложен файл пояснительной записки с обоснованием выбора технических решений: в первую очередь протоколов и форматов хранения ключа, шифр-текста. Объём пояснительной записки — достаточный для обоснования решений. Параметры криптосистем подбираются в процессе технического проектирования авторами исходя из эргономики и обеспечения стойкости в 80 битов. Выбор параметров обосновать в пояснительной записки.
2. Оценивается правильность работы программы (вклад в оценку — 50%) и её эргономика (вклад в оценку — 5%).
3. Плагиат кода обнуляет выполнение задачи. И она не засчитывается. Плагиатом не является любое заимствование с указанием его авторства. Заимствованный участок кода не включается в оценивание. Например, Вы не смогли реализовать функцию и заимствовали её у друга, указав это в программе. Реализация этой функции исключается из вклада в оценку правильности работы программы на основе (принципиальности) этой функции в программе. Важность вещь разумно-субъективная, определяемая лектором курса. Так, если Вы заимствовали реализацию криптосистемы. То, ясно, что это важная функция, т. к. она проверяет знания по теме курса. И её вклад может быть до 20% вклада кода программы в оценку. В этом случае, этот вклад вычитается из общего вклада в 50% и получается 30% вклада программы. Если же была заимствована функция реализации интерфейса командной строки, то она не относится к тем функциям, которые призваны проверить знания в области криптографии с открытым ключом, поэтому её принципиальность может быть оценена не более, чем в 5%.
4. Для получения оценки отлично нужно выполнить задание более, чем на 79%, оценка четыре ставится за выполнение на 65% — 79%, за 50% — 65% — тройка, и при выполнении задания менее, чем на 50% ставится — «неудовлетворительно».