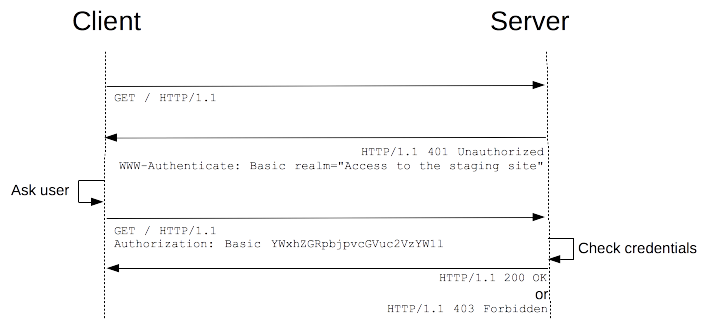
## Аутентификация в защищенном блокноте

Вариант 1. Выполнил Лобко Артем.

В качестве первого фактора механизма аутентификации был выбран **Basic access authentication**



**Server**

Когда сервер хочет, чтобы User Agent аутентифицировал себя по отношению к серверу, сервер должен соответствующим образом отвечать на неаутентифицированные запросы.

Для неавторизованных запросов сервер должен вернуть ответ, заголовок которого содержит статус HTTP 401 Unauthorized и поле WWW-Authenticate.

Поле WWW-Authenticate для базовой аутентификации построено следующим образом:

WWW-Authenticate: Basic realm="User Visible Realm"

**Client**

Когда пользовательский агент хочет отправить учетные данные для аутентификации на сервер, он может использовать поле авторизации.

Поле авторизации построено следующим образом:

1. Имя пользователя и пароль объединяются одним двоеточием (:). Это означает, что само имя пользователя не может содержать двоеточие.
2. Результирующая строка кодируется в последовательность октетов.
3. Результирующая строка кодируется с использованием варианта **Base64**.
4. Затем к закодированной строке добавляется метод авторизации и пробел (например, **«Basic»**).

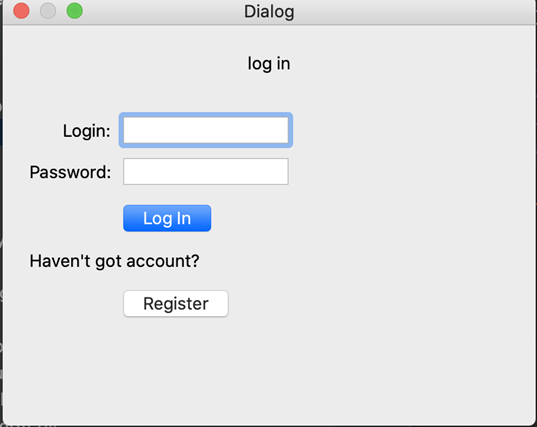
Например, если браузер использует Aladdin в качестве имени пользователя и OpenSesame в качестве пароля, то значение поля - это кодировка Base64 Aladdin: OpenSesame или QWxhZGRpbjpPcGVuU2VzYW1l. Тогда заголовок авторизации будет выглядеть так:

Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l

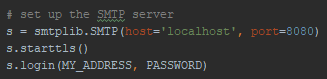
Собственно в **Basic access authentication** используется **base64** для создания токенов доступа.

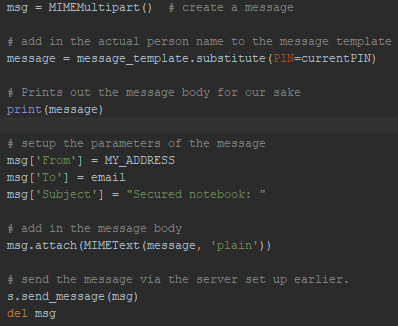
**base64** — это схема, по которой произвольная последовательность байт преобразуется в последовательность печатных ASCII символов. Используются только символы латинского алфавита в верхнем и нижнем регистре — символы (A—Z, a—z), цифры (0—9), и символы «+» и «/», с символом «=» в качестве специального кода суффикса. Полная спецификация этой формы base64 содержится в [RFC 1421](https://tools.ietf.org/html/rfc1421) и [RFC 2045](https://tools.ietf.org/html/rfc2045). Эта схема используется для кодирования последовательности [октетов](https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%82%D0%B5%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) ([байт](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82)).

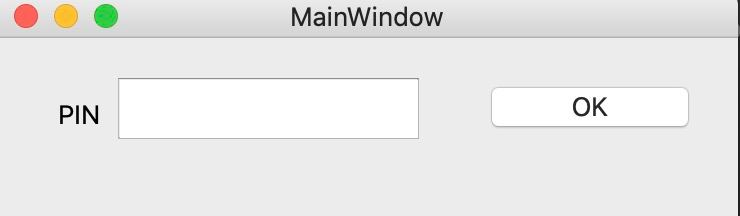
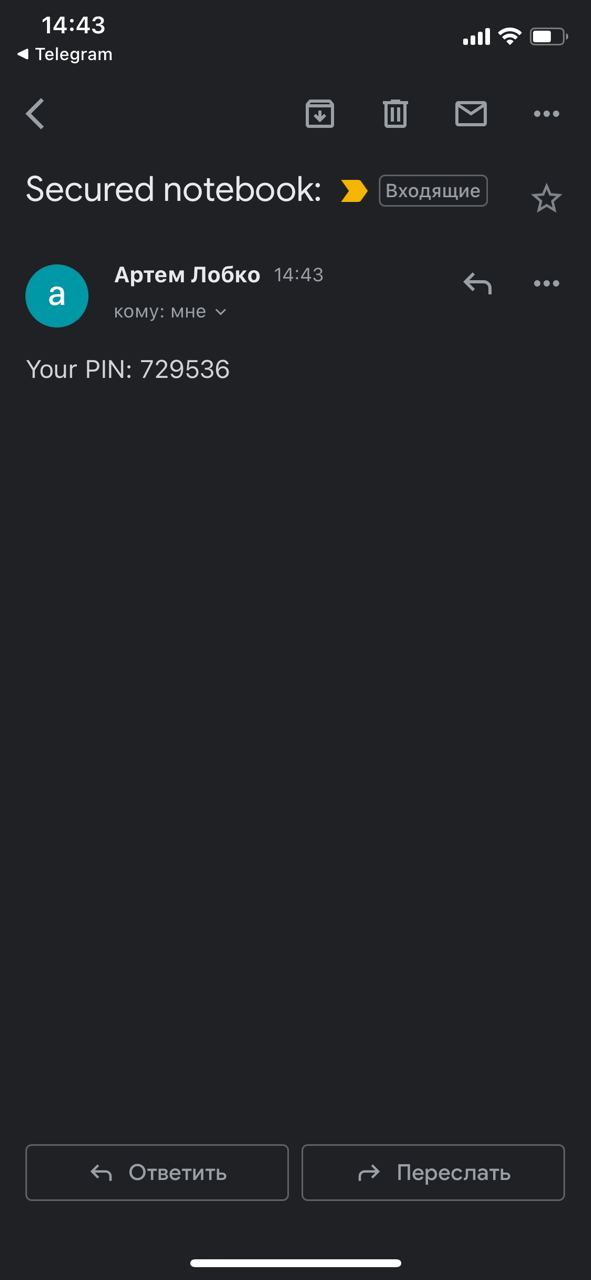
Для того, чтобы преобразовать данные в base64, первый байт помещается в самые старшие восемь бит 24-битного буфера, следующий — в средние восемь и третий — в младшие значащие восемь бит. Если кодируется менее, чем три байта, то соответствующие биты буфера устанавливаются в ноль. Далее каждые шесть бит буфера, начиная с самых старших, используются как индексы строки́ «ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/» и её символы, на которые указывают индексы, помещаются в выходную строку. Если кодируются только один или два байта, в результате получаются только первые два или три символа строки, а выходная строка дополняется двумя или одним символами «=». Это предотвращает добавление дополнительных битов к восстановленным данным. Процесс повторяется над оставшимися входными данными.



Вторым фактором аутентификации является отправка PIN-кода на почтовый ящик пользователя с периодом валидности в 2 минуты.







### Возможные атаки

Атака на систему аутентификации с base64 основана на подборе Authentication токена, что хорошо делается Broot Force’ом. С PIN’ом сложнее, так как для доступу к нему, злоумышленник должен иметь доступ к почтовому ящику жертвы. Получение доступа к которому может добавлять дополнительные факторы на пути к данным в защищенном блокноте.

Надо понимать, что base64 устарел и является небезопасным решением для аутентификации.

### По завершению процесса двухфакторной аутентификации, происходит обмен ключами для алгоритмов шифрования RSA, AES (по условию Лабораторной работы №2).

### 