УДК 519.813.3

Иванов И.С., Пьянков С.И., Сафина Л.Р.[1]

[1] Новосибирский государственный университет

Применение коммутирующих отображений в криптографии

1. Введение

Обозначим множество строк длины n над конечным алфавитом Ω через $S_n(\Omega)$. Множество всех взаимно однозначных отображений множества $S_n(\Omega)$ на себя будем обозначать через $Aut(S_n(\Omega))$. Элемент S из множества $Aut(S_n(\Omega))$ будем называть ключом, а действие S на элемент $u \in S_n(\Omega)$ назовем шифрованием. Пусть есть Алиса A и Боб B, и Алиса хочет послать Бобу некоторое сообщение $u \in S_n(\Omega)$ так, чтобы злоумышленник не догадался о его смысле. Тогда она берет некоторое отображение $T \in Aut_n(S_n(\Omega))$ и создает строку $v = Tu$, которую посылает Бобу. Бобу надо прочитать это сообщение, но он не ничего не знает об отображении T. Тогда он выбирает некоторое отображение $R \in Aut(S_n(\Omega))$ и возвращает Алисе строку $w - Rv - RTu$. Предположим, что выбранные отображения R, T - коммутирующие, т.е. имеет место равенство $RTu = TRu$. Тогда Алиса действует на строку w обратным отображением T^{-1} и отправляет Бобу. Боб получает строку $T^{-1}w = T^{-1}RTu = T^{-1}TRu = Ru$, после чего восстанавливает строку u, действуя обратным отображением R^{-1}.

Основная проблема данного метода возникает при построении коммутирующих отображений: множество строк над некоторым алфавитом имеет достаточно бедную структуру для построения коммутирующих отображений.

2. Методы линейной алгебры

Пусть есть векторное пространство F^n над конечным полем F. Будем рассматривать линейные отображения $A : F^n \to F^n$. Всякому линейному отображению соответствует матрица линейного отображения. Множество матриц размера $n \times n$ над полем F обозначим через $Mat_n(F)$. Два линейных отображения коммутируют тогда и

только тогда, когда матрицы этих отображений коммутируют. Пусть $A \in Mat_n(F)$ и пусть $G(x)$ - некоторый многочлен над полем F. Тогда для всякого такого многочлена определен матричный многочлен $G(A)$. Очевидно, что для всякого многочлена $G(x)$ матрица A и матрица $G(A)$ коммутируют. Кроме того, для всякой пары многочленов $G_1(x), G_2(x)$ матрицы $G_1(A), G_2(A)$ коммутируют.

Тогда построим следующий секретный протокол обмена сообщениями. Пусть Алиса хочет передать Бобу секретное сообщение. Алиса и Боб знают секретный ключ — случайную невырожденную матрицу $A \in Mat_n(F)$, и длину сообщения n. Они создают две обратимые матрицы как полиномы от матрицы A ($T = G_1(A)$, $R = G_2(A)$). Тогда, для обмена сообщением (вектор u длины n) выполняются следующие шаги:

Шаг 1 (Алиса): вычисление $v - Tu$, отправка v.

Шаг 2 (Боб): вычисление $w = RTu$, отправка w.

Шаг 3 (Алиса): вычисление $z = T^{-1}w = T^{-1}RTu = Ru$, отправка z.

Шаг 4 (Боб): вычисление $R^{-1}z = v$.

Отметим один нюанс: для обеспечения стойкости данного протокола необходимо регулярно менять секретный ключ, так как с ростом количества посланных сообщений количество неизвестных растёт медленнее числа уравнений, и с некоторого момента количество информации станет достаточным для криптоанализа и, возможно, дальнейшего взлома. В этом состоит теоретическая уязвимость алгоритма — при использовании постоянного секретного ключа количество неизвестных в перехваченных сообщениях растёт по формуле $n^2 + n * 3 * t$ (неизвестны матрица M размера $n \times n$ и три вектора длины n на каждое сообщение — коэффициенты полиномов и само сообщение), где t - количество переданных сообщений, а количество уравнений - $5 * n * t$. Таким образом, при $t > \frac{n}{2}$ количество уравнений превысит количество неизвестных и, с теоретической точки зрения, систему уравнений можно будет решить. На практике же это мало реализуемо, так как в итоговую систему уравнений входят уравнения $A_i x_i = u_i, B_i x_i = w_i$, которые не являются линейными относительно своих неизвестных, что, вкупе с работой над конечным объектом, не позволяет на современном уровне развития вычислительной техники за разумное время

вскрыть данную систему. Вскрытие также возможно при условии совпадении вектора сообщения и собственного вектора матрицы оператора. Для устранения данной уязвимости необходимо потребовать неприводимости характеристического полинома матрицы A над полем F.

3. Алгоритм идентификации

Предположим, что Алиса хочет доказать Бобу, что она действительно та, за которую себя выдает. Для идентификации у Алисы и Боба есть заранее оговоренные фиксированные вектора v_1, v_2 над заданным полем F. Считается, что вектора известны только им. Запишем алгоритм идентификации:

Первый шаг: Алиса создаёт A, $S = G(A)$ и посылает Бобу A, $w_1 = Sv_1$, $w_2 = Sv_2$. Боб по первому вектору определяет S, а по второму проверяет подлинность. Злоумышленник знает A, w_1, w_2, но никаким образом не может восстановить v_1, v_2.

Второй шаг: Боб создаёт два новых вектора v_1, v_2 и посылает Алисе новую пару $w_1 = Sv_1$, $w_2 = Sv_2$, после чего запоминает новый ключ идентификации Алисы. Алиса находит новые v_1, v_2, и также их запоминает для дальнейшего использования.

4. Игра в морской бой с проверкой честности

Предположим, что Алиса и Боб хотят провести игру в морской бой и после игры проверить честность оппонента. Для этого Алиса и Боб договариваются о секретном ключе — матрице M размера 13×13 над $GF(2^8)$. Далее Алиса и Боб придумывают коэффициенты полинома 12-й степени и применяют этот полином к матрице M, причём полученная матрица должна быть обратимой. Начало игры в морской бой представляет из себя размещение наборов единиц в нулевой матрице 10×10. Битовая матрица преобразуется в вектор из 13 байт разбиением матрицы на 8-элементные блоки слева направо и сверху вниз. После этого Алиса и Боб умножают матрицы полиномов на векторы игровых полей и отсылают друг другу. После того, как игра завершена, Алиса и Боб обмениваются матрицами, обратными к A и B соответственно, восстанавливают игровые поля оппонента и проверяют, совпало ли расшифрованное поле с тем, которое было открыто в процессе игры. Если в процессе игры оппонент жульничал (единственный способ сделать это в игре в морской бой — это передвигать свои корабли в ответ на ходы другого игрока) — поля не совпадут.