Основные атаки на алгоритм Эль Гамаля

Пока Алиса и Боб обменивались секретиками, Ева чувствовала себя брошенной. Она решила научиться читать сообщения друзей. Давайте поможем ей в этом интересном занятии!

Атака одинаковых сессионных ключей

Пока Боб ненадолго отлучился, хитрая Ева смогла подсмотреть в переписку. Она увидела то, что все сообщения шифруются при помощи одного и того же сессионного ключа. Кроме того, она смогла запомнить одно из сообщений.

Такого набора данных хватит, чтоб Ева спокойно могла прочитать любое сообщение, которое будет отправлено с данными параметрами.

Предположим, что при отправке двух разных сообщений m и m′ был использован один и тот же секретный ключ k. (Параметры p,g,y,x тоже одинаковые). Тогда рассмотрим шифртекст, который получается в таком случае:

$$m: u=g^k(modp),v=m\cdot(y^k)(modp)$$
$$m': u'=g^k(modp),v'=m'\cdot(y^k)(modp)$$
$$v'/v=m'/m\Rightarrow m'=v'\cdot m/v$$

Ура! При помощи такой формулы мы спокойно можем получить информацию о другом сообщении, которое было передано с таким же k.

Когда мы рассказали Еве о данной атаке, она была на седьмом небе от счастья. Но недолго она оставалась такой: Алиса и Боб догадались о том, что их сообщения были прочитаны, поэтому они устранили уязвимость: решили каждый раз генерировать случайный сессионный ключ. Но при этом они продолжили использовать относительно маленькие числа для ключей. Этим мы и воспользуемся!

Атака малого модуля

Атака состоит в том, что мы будем искать решение y=g^x(modp), зная y, g и p (помним, что параметры имеют относительно малое значение), алгоритмом baby step giant step.

Пусть есть уравнение y=g^x в цикличекой группе порядка n. Положим, что x=i·m−j, где m - заранее выбранная константа (обычно используют округленное вверх значение квадратного корня из p-1). Очевидно, что любое x из промежутка [0,p) можно представить в такой форме. Тогда уравение принимает вид: y=g^(m·i−j) ⇒y·(g^(−m))^i=g^j Алгоритм предварительно вычисляет g^i для некоторых i. Потом корректируется значение m и пробуются различные значения j.

Давайте рассмотрим пример:

$$20=5^x(\mathrm{mod}53)$$

В данном случае g=5, y=20 и p=53, и мы хотим узнать x. Для начала определим квадратный корень из p−1, и округлим до ближайшего целого:

$$m=\lfloor\sqrt{p-1}\rfloor=\lfloor\sqrt{52}\rfloor=7$$

Затем мы вычислим g^i(modp) от 1 до m и занесем информацию в виде словаря {gi(modp),i}

$$\{1:0,5:1,25:2,11:3,42:4,51:5,43:6,3:7\}$$

Например: Если i=6, мы получаем $5^6(\mathrm{mod}53)\equiv 43\Rightarrow\{43:6\}$
Теперь у нас есть список пар от 0 до квадратного корня из p−1. Вычислим g^(−m). Согласно одному из следствий малой теоремы Ферма g^(−m)=g^(m(p−2))=18 Затем мы проходимся по значениям y·(g^(−m))^j(modp)=20·18^j(mod53) пока мы не найдем совпадения в таблице. Тогда мы умножаем число на m и добавляем число, которое сопоставлено в таблице.

$$j=0:\ 20(\mathrm{mod}53)=20$$
$$j=1:\ 20\cdot 18(\mathrm{mod}53)=42$$

Число 42 есть в нашем словаре, значит $x=1*7+4=11$
Действительно, если выполнить проверку, то получится верное равенство!
Таким образом, мы знаем более эффективный алгоритм нахождения решения

уравнения y=g^x(modp), чем полный перебор.

Как можно избежать данной атаки? Во-первых, следует использовать достаточно большие значения модуля p (порядка 2^10 — 2^11 знаков). Во-вторых, нужно использовать числа Софи Жермен.

Простое число p является простым числом Софи Жермен, если 2p+1 также является простым.

Пример: 11 - простое число, 2*11+1 = 23 - связанное с ним простое, значит 11 - число Софи Жермен.

Первые несколько чисел Софи Жермен: 2, 3, 5, 11, 23, 29, 41, 53, 83, 89, 113, 131, 173, 179 ....

Числа 2p+1, объединенные простотой Софи Жермен, называются безопасными простыми. (Безопасное простое - это простое число вида 2p+1, где p - число Софи Жермен) Понятие безопасной простоты можно усилить до сильной простоты, для которой p-1 и p+1 имеют большие простые множители, которые, в свою очередь, имеют достаточно большие простые делители. (Сильное простое - безопасное простое число p, для которого p-1 и p+1 имеют большие простые делители, которые, в свою очередь, имеют достаточно большие простые делители)

Если для Z/pZ каноническое разложение числа p-1=2q (т.к p -простое, то p-1 четное), где q - безопасное или сильное простое число, то взлом криптосистемы Эль Гамаля методами "baby step", "baby step giant step" становится весьма затруднительными, практически невозможными.

Атака Meet in the Middle

Очередная атака, которую мы попробуем называется Meet in the Middle (встреча в середине). Не стоит путать эту атаку с Man in the Middle (человек по середине), о которой мы говорили в отдельном файле.
Если соблюдается ряд условий и сообщение m, которое Боб посылает Алисе, достаточно короткое, то Ева может восстановить m, если она сможет узнать v (а она сможет, т.к это открытая информация), когда Боб посылает свой шифротекст Алисе. В частности, Meet in the Middle работает только в том случае, если:

- m состоит из B бит, причем Ева знает B, и B является небольшим числом.
- m можно разделить на две части таким образом, что m=(m1)(m2). Пусть m1 состоит из битов b1, а m2 - из битов b2.
- Ева знает n. Поскольку сессионный ключ Боба k обладает свойством $1 \leq x \leq n-1$, ему также необходимо каким-то образом определить n. Если он может, то Ева может тоже.
- m не является элементом подгруппы, порожденной g. Это кажется совершенно маловероятным. Однако, порядок n подгруппы, генерируемой g, часто выбирается небольшим по соображениям эффективности.
- Порядок n подгруппы, порожденной g, имеет свойство, что $n \leq (p-1) \cdot 2^{(-b)}$. Опять же, возможно, что n мало по соображениям эффективности.

Если все магическим образом будут выполнены все условия,то Ева может действовать следующем образом:
Она знает, что

$$y=g^x(\mathrm{mod}p)$$

и что

$$v=m \cdot (y^k)(\mathrm{mod}p)$$

Возведем обе части в степень n:

$$v^n=(m^n) \cdot (y^{(nr)})(\mathrm{mod}p)$$

Это бесполезно как часть атаки, если m является элементом подгруппы, порожденной g, потому что тогда $m^n=1$ (поскольку все элементы подгруппы генерируют эту подгруппу, хотя и в другом порядке)

Однако если это не так, вспомним, что gn=g0=1, поскольку подгруппа, порожденная g, циклична, так что

$$y^{kn}=g^{xkn}=(g^n)^{xk}=1^{xk}=1$$

Получаем, что

$$v^n=m^n(modp)$$

Используя предположение, что m=(m1)(m2):

$$v^n=(m1^n)(m2^n)(modp)$$
$$(v^n)(m2^{(-n)})=m1^n(modp)$$

Действовать мы можем также, как и в алгоритме baby step giant step, т.е генрировать словарь {m1^n(modp);m1} для всех m1, а для всех m2 вычислять выражение (v^n)(m2^(-n))(modp) и искать его в словаре. Если мы найдем совпадение в словаре, мы нашли решение (m1,m2) для (v^n)(m2^-n)=m1^n(modp) и m *возможно* равно (m1)(m2).

Атаки на алгоритм цифровой подписи

Атака на плохо выбранные разовые ключи

При рассмотрении стойкости необходимо соотношение m≡xr+ks(modp−1). Тот, кто наблюдает за подписывающим, видит серию подписанных документов:

$$[m1,r1,s1]\longrightarrow m1\equiv xr1+k1s1(modp-1)$$
$$[m2,r2,s2]\longrightarrow m2\equiv xr2+k2s2(modp-1)$$
$$\dots$$
$$[mn,rn,sn]\longrightarrow mn\equiv xrn+k1sn(modp-1)$$

Неизвестные тут: x,k1…kn. Получили систему уравнений от n+1 неизвестных, т.е. неопределенная система уравнений. Если знаем хоть один разовый ключ, то система становится определенной и решается - значит, все ключи должны быть случайными, секретными и разными.

Атака "по корректной тройке"

Атака связана с построением цифровой подписи по известной корректной тройке [m,r,s]. Если взять параметры A,B,C такие, что ∃(Ar−Cs)^(−1)(modp−1) то можно построить целую серию документов, удовлетворяющих соотношению проверки.

$$r' = r\text{^}A*g\text{^}B*y\text{^}C(modp)$$
$$s' = s\cdot r'/(Ar-Cs)\text{^}(-1)(modp-1)$$
$$m' = r'(Am-Bs)/(Ar-Cs)(modp-1)$$

Создав таким образом множество документов, можно организовать DDOS-атаку на проверяющего. Защита: m′ - случайное число, m=Hash(m′) - значит, цифровая подпись для Эль Гамаля должна применяться вместе с Хеш-функцией.