Advanced Cryptography

Guy Coop - gtc434 - 1447634

January 28, 2018

Question 1

Using the Pohlig-Helman Algorithm to solve discrete log
given $h = g^x \; mod(p)$
where: $p = 31$, $g = 3$, $h = 13$
find the value of x ...

$$N = p - 1 = 30 = 2^1 \cdot 3^1 \cdot 5^1$$

q	e	$g^{(p-1)/q^e}$	$h^{(p-1)/q^e}$	solve $(g^{(p-1)/q^e})^x = h^{(p-1)/q^e}$ for x
2	1	30	30	$30^x = 30 \rightarrow x = 1$
3	1	25	5	$25^x = 5(mod31) \rightarrow x = 2$
5	1	16	16	$16^x = 16 \rightarrow x = 1$

$$x = 1 \; mod(2) = 2 \; mod(3) = 1 \; mod(5)$$

using chinese remainder theorem ...

$x = 5j + 1$ for some integer j
$5j + 1 = 2 \; mod(3) \rightarrow j = 2 \; mod(3)$
$j = 3k + 2$ for some integer k
$x = 5(3k + 2) + 1 = 15k + 11$
$15k + 11 = 1 \; mod(2) \rightarrow k = 0 \; mod(2)$
$k = 2l$ for some integer l
$x = 30l + 11$
therefore:

$$x = 11 \; mod(30)$$

to confirm this is the correct answer we can subsitute it back into the equation:

$$3^{11} \; mod(31) = 13$$

Question 2

part a) discuss the main rational for cryptographic key sizes

Cryptographic key sizes are typically calculated by the some function of the level of security you wish to provide. Informally, the key sizes is selected such that an attacker with X resources will take Y amount of time to break the cryptography where X is selected by the user and Y is some arbitrarily large amount of time, that can be considered too long to reasonably attempt. But as key sizes increase there is a drop in throughput on the user side, so while a "long" key would take considerably longer to break, it would also add more overhead to each user operation than a "short key". The optimal method therefore is to decide the maximum resource that you would ever anticipate needing to protect against and then select a key length that will be just adequate for that level of resource.

part b) What are the key length recommendations for security based on factoring in 2050

Reccomendation provider	Minimum key length recommendation
Lenstra / Verheul	4047
Lenstra updated	2440
ECRYPT II	15424
NIST	7680
ANSSI	3072
IAD-NSA	3072
BSI	3000

part c) If there are any differences between the various recommendations provided, how could these differences be explained?

The differences in these recommendations can be partially explained by the large time gap between their dates of publication. For instance Lenstra/Verheul was published in 2000 compared to NIST in 2016. Additionally, it appears likely that the recommendations anticipate different levels of adversary, based on their corresponding recommendations for symetric key cryptography which range from 102 - 256 bits. Finally, the older algorithms specifically, account for Moor's law, but do not account for the sudden jump in computational efficiency (specifically for the factoring problem) that could occur with the introduction of quantum computers, Assuming that Shor's algorithm does not apply, otherwise factoring based cryptography is no longer a viable choice.

Question 3

From the description in "A one round protocol for tripartite Diffie-Hellman" by Antoine Joux:
Given two points on the curve P and Q, participants A, B, and C choose three random values a, b, and c respectively. each participant then computes and transmits $([a]P, [a]Q)$, $([b]P, [b]Q)$, and $([c]P, [c]Q)$ Using these values, A, B, and C can respectively compute the common key as:
$F_T(a, (P_B) - (Q_B), (P_C + Q_C) - (O))$,
$F_T(b, (P_A) - (Q_A), (P_C + Q_C) - (O))$,
$F_T(c, (P_B) - (Q_B), (P_A + Q_A) - (O))$
where F_T is a Tate pairing