

RE: Encryption question

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/security-basics/2004-02/0414.html>

From: Prasad S. Athawale (*athawale_at_cse.Buffalo.EDU*)

Date: 02/26/04

To: "'Jordan, Jason D. \"Dallas\"'" <Jason.Jordan@honeywell-tsi.com>, "'Preston, Tony'" <Tony.Pre
Date: Wed, 25 Feb 2004 23:45:19 -0500

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Hi!

The public/private key pair is generated together at the same time – and one cannot at least it's termed 'mathematically difficult' determine the other key when one key is known – which is the essence of public-key cryptography.

So when you want to generate another private key – to be used with Alice's public it effectively has to Alice's private key that you somehow generate – or else the decryption just wouldn't work. And this like I said earlier is mathematically difficult.

As regards the man-in-the middle attack – this can be done like what Jason said – about suppressing the knowledge of Alice's 'true' public key and providing Bob with your own fake 'Alice's public key. This fake key would be the public key from a public-private key pair that you generated. So whenever Bob encrypts something with the public key of Alice in his possession, it effectively can be decrypted by you (if you can intercept the message of course), re-encrypted with Alice's true public key – which you possess and forwarded on to Alice. Alice would in turn be able to read the message because it is encrypted with her public key.

Here's the relatively easy to understand math behind RSA public key encryption.

1. Find P and Q, two large (e.g., 1024-bit) prime numbers.

SecurityFocus BASICS: RE: Encryption question

2. Choose E such that E is greater than 1, E is less than PQ , and E and $(P-1)(Q-1)$ are relatively prime, which means they have no prime factors in common. E does not have to be prime, but it must be odd. $(P-1)(Q-1)$ can't be prime because it's an even number.
3. Compute D such that $(DE - 1)$ is evenly divisible by $(P-1)(Q-1)$. Mathematicians write this as $DE = 1 \pmod{(P-1)(Q-1)}$, and they call D the multiplicative inverse of E . This is easy to do — simply find an integer X which causes $D = (X(P-1)(Q-1) + 1)/E$ to be an integer, then use that value of D .
4. The encryption function is $C = (T^E) \pmod{PQ}$, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and $^$ indicates exponentiation. The message being encrypted, T , must be less than the modulus, PQ .
5. The decryption function is $T = (C^D) \pmod{PQ}$, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and $^$ indicates exponentiation.

Your public key is the pair (PQ, E) . Your private key is the number D (reveal it to no one). The product PQ is the modulus (often called N in the literature). E is the public exponent. D is the secret exponent.

The difficult part here is determining $D=E^{-1} \pmod{(P-1)(Q-1)}$, when P & Q are unknown — you just know their product.

Also since someone mentioned about the primality of a number — it's a known difficult problem in mathematics a solution to which has been recently proposed.

<http://mathworld.wolfram.com/news/2002-08-07/primetest/>

And remember PKE is based on the premise — that everyone can keep their private key secret, and the authentic public key's are readily and reliably available to everyone from an authentic server.

Hope this helped,

Prasad

— -----Original Message-----

From: Jordan, Jason D. "Dallas"
[mailto:Jason.Jordan@honeywell-tsi.com]
Sent: Wednesday, February 25, 2004 12:45 PM
To: 'Preston, Tony'; 'security-basics@securityfocus.com'
Subject: RE: Encryption question

Tony,

RE: Encryption question

SecurityFocus BASICS: RE: Encryption question

I believe, in order to spoof a digital signature of Alice, you would need to get her private key....which she should have securely stored somewhere. A hash of the message is done and then encrypted with Alices private key. The only other key that

can decrypt it is the public key generated with her original key pair. You could substitute Alice's public key with your public key so when Bob used that public key to encrypt the message meant for Alice, you could intercept it and read the message.

Then you could re-encrypt it with Alice's real public key and send it on to her. Kinda like a man in the middle deal. I think this is how it works, I could be wrong. Does that help any?

Dallas Jordan MCSE, CCNA, Security+

Electronics Technician II

Honeywell Technology Solutions

1010 Bankton Drive

Hanahan, SC 29406

843-744-1221 Ext 11

-----Original Message-----

From: Preston, Tony [mailto:Tony.Preston@acs-inc.com]

Sent: Tuesday, February 24, 2004 1:01 PM

To: security-basics@securityfocus.com

Subject: Encryption question

Tony Preston

Systems Engineer, AS&T Inc.

RE: Encryption question

SecurityFocus BASICS: RE: Encryption question

Division of L3 Corporation

(609) 485-0205 x 181

I have what is a rather basic question... I probably am missing something

so I thought I would ask here.

Alice and Bob both have a public and private key.

Alice encrypts her email to Bob using his public key. Sends the email and

Bob decrypts it using his keys..

Since both Bob and Alice's public keys are known, Why can't I take Alice's

public key and create a key pair using any other private key. Now, I fake

an electronic signature from Alice using the pair I created and send a bogus

encrypted message to Bob with my "fake" Alice signature. Bob checks the

signature by using the public key and it is valid. Bob assumes the message

is from Alice...

What prevents me from spoofing someone's electronic signature this way?

SecurityFocus BASICS: RE: Encryption question

-----BEGIN PGP SIGNATURE-----

Version: PGPfreeware 6.5.8 for non-commercial use <<http://www.pgp.com>>

iQA/AwUBQD153oKN2ncVpx7SEQK3JgCfZQxWLVKt6VwQPf3xvQK0fv8pH0MAniet
dcrm/tWbvhSf/gY3JxLbmwY0
=1Bty

-----END PGP SIGNATURE-----

