algo^{logic}
Algologic Research and Solutions
Technical Report

Negotiating public keys in asymmetric key cryptography

Dec. 2017

Algologic Technical Report
Dec. 2017

Negotiating public keys in asymmetric key cryptography

S. Parthasarathy
drpartha@gmail.com

Our Ref.: publikey.tex
Date: 2017-12-31a

Executive summary[1][2]

Asymmetric key cryptography involves using a pair of keys for encrypting and decrypting a message. It avoids the risk involved in tranferring the key, as in the case of symmetric key crytography. However, asymmetric key cryptography has a vulnerability, known as the man-in-the middle attack. This report proposes a protocol which will tackle the problem of man-in-the middle attack.

Table of contents

Contents

1 Asymmetric key cryptography **3**

2 Enter Sita and Rama, exit Alice and Bob **4**

3 Man in the middle **5**

[1] You can get the LaTeX source (Ref.: publikey.tex, Sep. 2012), and the rendered version, from drpartha@gmail.com. This LaTeX based report was prepared using Kile, on a Suse Linux system.

[2] This report can be downloaded from http://profpartha.webs.com/publications/publikey.pdf

4 Public key negotiating protocol **6**
4.1 Part 1 6
4.2 Part 2 6

5 Concluding remarks **7**

6 About the author **8**

1 Asymmetric key cryptography

Cryptography is the art of scrambling up a message or piece of information, such that an unauthorised person cannot read or use the message/information. Cryptography also involves de-scrambling the message to make it legible. Good cryptography ensures that the descrambling can be done only by the person who is authorised to descramble. Cryptography is useful whenever a message/information is confidential, and needs to be protected against unauthorised reading or modifications. The scrambled message is called the *cipher text*, and the descrambled message is called the *clear text*. The process used for scrambling/encryption and descrambling/decryption is called a *cryptographic scheme*.

Cryptographic schemes usually consist of two distinct components : a *cryptographic key*, and a *cryptographic algorithm*. The algorithm may be open and known, but the key is kept confidential by the person who uses the cryptographic scheme. All the secret is in the key. Here is a small example (also called the Caesar's cipher) :

Name : Caesar's cipher
Key : 3
Algorithm : Shift every character forward by the number indicated by the key (in our case 3)

Clear text : PARTHA
Cipher text : TEVXLE

Given the cipher text TEVXLE, we need to know the key (3) and the algorithm used for generating the cipher. It is now an easy work back, to generate the clear text. Notice that we need the same key (3) for encryption as well as decryption. This scheme is called as *symmetric key encryption*. **The receiver must know this key, if he/she has to decrypt a given message**. This requirement poses a certain risk. If the message can be stolen, so can the key be too ! It is not enough if the receiver knows the algorithm, he must know the key also. Transferring/sharing the key is the most sensitive part of such encryption mechanisms.

The problem of key transfer/key sharing, is avoided altogether in *Asymmetric key encryption*. Asymmetric key encryption, uses two keys. One key (called the public key) is used for encryption. A different key (called the private key) is used for decryption. The public key can be visible to anybody

who wants to encrypt a message. The private key (corresponding to a given public key) is known only to the receiver of the message. The problem of key transfer/key sharing, is avoided altogether. Only the authorised receiver (who knows the private key) will be able to decrypt a message which was earlier encrypted using his public key.

The person who wishes to receive encrypted messages, creates the key pair. He retains his private key safely, and distributes his public key.

Public keys can be distributed in various ways :

- The creator of the key can send it to his correspondants, directly, on a one-to-one basis. If the correspondant already has a public key, the creator will encrypt his public key and send to the correspondant (who can then decrypt the message, and retrieve the key)
- The creator of the key can store his key in any of the public key servers. Anyone can retrieve the key from the server
- The creator of the key can put up the public key on his own web server

2 Enter Sita and Rama, exit Alice and Bob

Traditionally, all books and documents about cryptography use the characters Alice and Bob : *Alice wants to send a message to Bob*

We use the Hindu mythological characters Sita and her husband Rama : *Sita wants to send a message to Rama*[3]. The choice is meaningful since the first letters of these names also indicate their role (s - sender, r - receiver). Alice and Bob can now stop sending messages to each other, and can go and enjoy a holiday.

We will also be talking about public keys and private keys. We will us the last letter C (for public) and E for (private) to identify these keys. Thus the public key of Rama will be C_{rama}, and the private key of Rama will be E_{rama}.

When Sita wants to send a secret message to Rama, Sita will use C_{rama} and encrypt the mesage. Rama will use his E_{rama} to decrypt the mesasge. Only Rama knows his E_{rama}. Before doing that, Sita must make sure that she is indeed using the authentic public key of Rama.

[3]This choice is inspired from an episode in the Hindu epic Ramayana

3 Man in the middle

Although asymmetric key cryptography solves the problem of key transfer from the sender to the receiver, it has a certain weak point. This is commonly known as the *man in the middle* attack. Imagine an unscrupulous person who replaces the public key by a spurious public key of his own (he has the corresponding secret key with him). The unsuspecting Sita uses this spurious key and encrypts his/her message. Two things can happen:

- Rama will never be able to decrypt the message using his own secret key E_{rama}. Thus the man-in-the-middle can block all secret messages from reaching Rama. Thus, important and strategic messages will also get blocked.
- The man-in-the-middle will be able to decrypt all the messages (which were encrypted with his spurious key), using his own private key. The confidentiality of data is seriously compromised.

It is thus important to counteract the threat caused by the man-in-the-middle. It is important for the sender and the receiver to know that the authentic public key has been replaced by a spurious key.

There exist some simple precautions which Sita can take.

- Before she uses C_{rama}, Sita will verify the *fingerprint* of the public key he/she has received, and confirm with Rama. But, this has itw own set of risks.
- In addition to this, Rama can also create and distribute the md5 sum of his public key. The man-in-the-middle will now have to replace the fingerprint, and the md5 sum also, in addition to replacing the public key.
- In addition to all this, Rama should keep checking periodically, all locations where he has kept his key for public access, and ensure that the keys have not been altered/replaced. It is the creator's moral responsibility to ensure that his public key is never misused.

To make the work of the man-in-the-middle even more difficult, the following protocol is proposed.

4 Public key negotiating protocol

This protocol must be applied, the very first time you use someone else's public key (as a sender). Sita must be she sure that she is using a genuine public key of Rama. Rama must know that he is using a genuine public key of Sita.

4.1 Part 1

- Sita will download the public key C_{rama} from a location indicated by Rama
- Sita will encrypt this key C_{rama} using the same public key C_{rama}
- Sita will send it to Rama

If the key is the genuine and authentic key, Rama will be able to decrypt Sita's mesage and get the key. He can compute the fingerprint of this key, and confirm that it matches the value he has with him ever since he created the key.

If Rama encounters a problem in decrypting the message, he can conclude that the public key used for encryption is not the authentic one. He will alert Sita, and take necessary precautions.

As soon as Rama realises an attempt to steal his C_{rama}, he can take two steps ::

- Revoke C_{rama}, and
- generate a new key pair

Now that Sita is sure that she has Rama's public key, she will have to help Rama get her public key safely.

4.2 Part 2

Sita will ::

- She will send Rama, her own public key, encrypted using Rama's public key. She has confirmed in Step 1 that she has indeed a genuine public key of Rama. Rama will be able to decrypt the message sent by Sita, and recover Sita's public key.

The public key negotiation protocol described above, if applied scrupulously, can nullify man-in-the-middle attacks.

- This protocol needs to be applied only once, i.e. at the time of first usage of a public key. Once the test is successful, you can keep the public key safely, and inaccessible to any man-in-the-middle. It is important to take all precautions, whenever security is involved.
- This protocol does not employ any other method or procedure, other than asymmetric key crytography. Thus no additional threats are likely to creep in.
- This protocol must be applied *in addition to* all the other usual precautions e.g. confirming te key fingerprint, confirming the md5 checksum, confirming the creator's identity etc.
- This protocol does not need the usage of any propreitary software or methods.
- This protocol involves only the senders and the receiver. It does not seek the assistance of an external agent like a Certificate Authority (used in Public Key Cryptography).
- This protocol requires close collaboration between the owner of the keys and the users. If there is a large number of users, this protocol can be cumbersome for the owner of the key. However, the effort involved is worthwhile, since it involves security.

5 Concluding remarks

Although asymmetric key crytography is generally superior to symmetric key crytography, in terms of dependability, it is vulnerable to man-in-the-middle attacks. This report proposes a protocol to protect onself from man-in-the-middle attacks.

The author of this report is a serious user of tools for security enhnancement. He uses a combination of GPG and md5 regularly. His GPG public key is accessible from:: http://drpartha.org.in/drpartha/publikey.htm

The author invites and welcomes suggestions from the reader.

6 About the author

Figure 1: The pensive Professor

S. Parthasarathy teaches discrete mathematics, and preaches LaTeX and Linux, to undergaduate students of Computer Science, in India. His association with LaTeX started in 1993, at the United Nations University, International Institute of Software Technology (UNU/IIST), Macau. One of the things he brought back from UNU/IIST was an unbounded admiration for Unix and LaTeX, both of which were used extensively at UNU/IIST. With the easy availability of Linux and front-end tools Kile, his conversion to Latex on Linux platforms was, more or less, inevitable. For more details about Dr. Partha, please visit :
`http://drpartha.org.in/`

You can interact with him, by e-mail, on :
`drpartha@gmail.com`

Index

About the author, 7
Asymmetric key cryptography, 2

Executive Summary, 1

Man in the middle, 4

Public key negotiation protocol, 5