public key cryptography

-- WHY IT DON'T WORK AND WHY YOU DON'T NEED IT

This report discusses fundamental issues on how public key encryption and public key infrastructure is implemented. The report concludes that public key encryption might be interesting mathematics, but practically no advantage is gained in its use. The report show that, if you set up a secure network, a single secure key transfer for each node is necessary, and classical symmetrical encryption can also be implemented using a single secure key transfer for each node. The discussion is limited to key distribution issues.

Symmetrical vs. Public Key Encryption

Encryption of a communication link using symmetrical encryption, where both receiver and sender share the same secret cipher key, is the classical case. Key distribution is performed by secretly carrying the key from node A to B using secure means. This is done for all pairs of communicating nodes. Distributed keys are usually replaced regularly or on demand, again using a courier that physically carries the keys between the nodes.

In public key encryption, the encryption is asymmetrical, and the encryption key is different from the decryption key. Only the decryption key needs to be kept secret. The advantage using public key cryptography is the possibility of exchanging keys in public, on the unsecure network, without the cost of using a courier.

In the man-in-the middle attack, the Opposition intercepts the transmitted public keys, and exchange them with his own, with the consequence that he can read and modify all communication to a particular node. This is why all public keys are signed by a Key Signing Authority. The public key of the key signing authority is published, and known by the nodes.

The Need for Secure Computation

In symmetrical encryption, we directly see the need for secure storage of the secret key, with the corresponding means to perform protected calculations, which perform the encryption functions for the node. In public key cryptography, which is in mathematical form, the need is not that obvious. But if an Opponent can influence, or control, the operation of the cipher, public key cryptography cannot work. The Opponent might simply enter any message to the node, and then influence the public key encryption unit to neither decrypt nor even check the message, it is simply forwarded with an approval. We conclude that when we use public key encryption, we still need a green box that looks like a mini-safe, where the public key encryption takes place. If you do not have it, you do not have any security!

Using a Key Server for Symmetrical Encryption

A key server, for symmetrical encryption, is a device that generates and distributes symmetrical keys. A node is introduced in the network by transferring a symmetrical key from the key server to the node. This key is shared by no one else. When communication shall be established, the key server generates a fresh key, using a true random number generator, and encrypts the key with secret keys corresponding to respective nodes. Each node can decrypt the communication key, which is used to secure the communication. When the message is sent, or the line dropped, the temporary key is discarded.
A typical case is where the key server also issues the equipment for the nodes, with the advantage that the keys can be loaded into the equipment before it ships. Obviously must the shipping take place using secure means, but this might be required for the equipment anyway, to prevent tampering.

The advantage with a key server; you need to transfer a single key only once between the server and a node. We can compare symmetrical and public key encryption, where we see that the key signing authority works as a key server, receiving public keys and returning the corresponding certificates.

Key Distribution Problems

For any kind of security system, that uses encryption, there is a general rule:
(theorem)

> You cannot introduce or improve
> security by encryption
> - you can only amplify the security
> already present in the key
> distribution channel.

This holds for both symmetrical and asymmetrical encryption. It can be seen as an axiom for encryption systems. We note that in asymmetrical encryption, where the security is asymmetrical, we can have different properties of the communication channel in different directions; it is possible to have authentication without secrecy, and vice versa. For a properly implemented secret key channel, we normally implement both secrecy and authentication at the same time, and usually we have separate secret keys for transmitting and receiving.

The problem is that in public key encryption, you do not have a well-defined key distribution channel. It is simply assumed that "it works". If you do not have a key channel, that provides security, this cannot later be implemented by encryption.

For a secret key system, we have seen that the key exchange between the key distribution centre and a node is the important key exchange. The security in the exchange is amplified into secondary keys and protocols that encrypt the messages. For public key encryption, the important step is key transfer to the key signing authority, where the public key is signed.

The certificate is checked using the public key of the key signing authority. But where does that come from? Has it been exchanged using secure means; secure with the quality that your network requires?

On Publishing the Keys

The dangerous part in public key cryptography is when you "publish" the keys. It is the concept of "publishing" that guide us astray. In the old days there where newspapers, that contained articles that where published. It would have been rather problematic and messy for the Opposition to interfere with the printing of the newspaper, and modify or add material.

We can summarise the properties of the printed article:
1) The Author can write an article, but no one else on the network can. All other nodes can only read the article.
2) The Author can be identified, and this identification is checked so that there is no error. The Opposition cannot mimic or add false published messages.
3) All nodes will receive the same copy of the article, and the Opposition cannot provide specific nodes with false articles

Together this is exactly the properties that we need for a secure key exchange using public key cryptography. The problem is that, as all information nowadays is carried electronically, the "Publish" on paper do not exist anymore. In fact, I doubt that any public key has ever been published, have you seen any? Can you find a published book of public keys in your local library?

One major error, in public key cryptography, is simply assuming that this works, instead of enforcing the rules by exchanging keys. What is needed is a courier who receives the public key of the key signing authority in person on site, and then travels to the node and installs a correct version in a key storage facility. In practice the public keys of the key signing authority are simply provided with the distribution of Microsoft Windows; with the reference to the theorem above, without a secure key distribution you can have no security!

Another problem is the integrity of these key-signing authorities – see the long list you have on your computer (certificate issuers that you "trust") – that it can be suspected that two or three of these cooperate with "National Security" and may sign false keys.

Carrying a Public Key

We conclude that, to provide good security that can be verified, we will need to carry keys physically also when using public key cryptography. Often, these keys are non-secret, meaning that no inconvenience occurs should the Opposition copy the keys in transit. He may, however, not be allowed to replace the keys in transfer, because then security is lost. We note that, to implement a secure key exchange by a currier, there is no advantage letting the Opposition copying the public keys. We must transfer the keys in a secure letter, possibly sealed or contained in a portable vault.

We conclude that if you need security, in public key cryptography, you must carry a key secretly and securely identically to when using ordinary symmetrical encryption. Due to this, nothing is gained with public key cryptography and it merely increase costs with no benefit.

Key Revocation Problems

Sometimes we need to revoke a key. A typical scenario would be that node "W" no longer works towards the assigned goal, so we need to cut node W out from the network. In a symmetrical encryption scheme, we simply need to inform the key server that it should no longer send out keys to node W. This is an on-line operation, which will likely take effect immediately. We note that node W don't have any parallel or secondary keys to the other nodes on the network, and as these nodes only accept communication that is decrypted using keys from the key server, no none on our network can be reached by W.

In public key cryptography, a message is accepted as genuine if the corresponding public key is signed. If we want to revoke a key, we need means to un-sign a public key. Clearly, some kind of register is needed where a node can check if a message from node W is valid, or if the keys have been revoked. We note that we must check the keys on-line, and, in practice, the register functions as a key server.

Compromised Nodes and Keys

A concept that we have in encrypted networks is a function where we can simply declare a node or office as invalid, by saying that it has been compromised. It should not be assumed that there are any facts or evidence; it is simply that we now have lost confidence in Bill in the computer centre, and we simply reassign him and his Coca Cola cans to some other work elsewhere. However, the computer centre is Compromised: we check all hardware for any modifications, and we replace all keys.

In classical symmetrical encryption, the keys are regularly replaced. This is not due to any cryptographic weakness in key distribution, implementation, algorithms, or maintenance, it is due to that the cryptographic keys may be lost due to other causes, such as an insider sells the keys or the Opponent simply demands the keys using treats or blackmail ("Practical" cryptanalysis). Obviously, also public keys need to be regularly replaced.

The problem with public key cryptography is that the whole system is not under control of those who run the network. In particular, a key signing authority cannot be declared as "Compromised". I have a feeling that no one have thought this over properly. Suppose that FBI pull out someone in handcuffs from the key signing authority. How do we proceed to invalidate associated keys and generate new ones?

How to Implement Complex Protocols

In public key cryptography, research is investigating and proposing new advanced methods for voting or information exchange or possibly monetary systems, using public key cryptography to enforce various advanced functions.
We note that, in classical symmetrical encryption, we have access to a green box that run the encryption, witch provide a secure environment. Therefore, we can simply include all kinds of advanced functions inside the encryption box, and we can implement this using ordinary software. This software needs to be high quality, and we must verify all functions of the software, but I promise you, we can write correct software and load it into a box if we need to.

Conclusions

I have shown that if you prefer public key cryptography, and if you need security, you must carry keys identically to when you use symmetrical encryption. In both cases a single key transfer for each node is enough too solve the key distribution problem. Consider changing the keys regularly.

We have also shown that all advanced properties of public key cryptography can also be implemented in a network using only symmetrical encryption. In addition to any protocol you fancy, you can easy add application specific restrictions, and enforce these with encryption.

Bo Dömstedt
TRNG98
E-mail: Bo Dömstedt <encryption@trng98.se>
www.TRNG98.se