System Security, Spring 2010 - Exercise 1

Exercise 1: Cryptography Basics

a) The explanation of the following terms is based on the sources: [3, 4]

   i) Integrity is the property that data has neither accidentally nor maliciously been modified or destroyed.

   ii) Confidentiality is the property that access to data is limited to individuals or processes that are authorized.

   iii) Denial of service is the prevention of authorized access to a system resource or the delaying of system operations.

   iv) Authentication is the confirmation of the claimed identity or origin.

   v) Authorization is the granting of the right to do something.

b) In symmetric cryptography the same key is used to encrypt and decrypt whereas in asymmetric cryptography the keys are different. Asymmetric keys have to be longer than symmetric keys since the mathematical problems that are used in asymmetric cryptography can usually be solved faster than brute forcing a symmetric key. And since asymmetric cryptography is computationally more expensive hybrid systems that use asymmetric as well as symmetric cryptography have to be used if a larger amount of data has to be encrypted.

c) Two possible applications for public-key cryptography are:

   a) The encryption and decryption of messages between two identities.

   b) The verification of a message by means of a digital signature.

d) In (public key) encryption of e–mails the receiver's public key is used to encrypt the message which then can be decrypted by the receiver's private key. Therefore we get confidentiality since the receiver is the only one who will be able to decrypt the message.

   When signing an e–mail the private key is used and everyone else who has the public key can verify the signature. We get the authentication that the message has been signed by the sender who is in possession of the private key.

e) The description of these attacks is based on the book [4]

   i) Ciphertext only: The attacker has only access to the ciphertext to be decoded. A typical approach would be brute–force where the attacker tries all possible keys.

   ii) Known plaintext: The attacker does not only have access to the ciphertext to be decoded but also to some plaintext–ciphertext pairs that have been produced with the secret key. On the basis of this knowledge the attacker may be able to generate a codebook or to deduce the key by analysing the way in which the known plaintext is transformed.

iii) Chosen plaintext: Instead of "random" plaintext–ciphertext pairs the attacker will now have pairs where he was able to choose the plaintext message. The attacker will pick plaintext–messages which contain patterns that are expected to reveal the structure of the key.

iv) Chosen ciphertext: In the chosen ciphertext attack, the attacker has again plaintext–ciphertext pairs, but instead of picking plaintext messages he has the possibility to choose the ciphertext. The attacker will again deliberately pick patterns so that the structure of the key will be revealed.

v) Chosen cipher– and plaintext: This attack is a combination of the chosen plaintext and chosen ciphertext attack. The attacker has plaintext–ciphertext pairs where he can choose the plaintext or the ciphertext and then gets the corresponding answer.

The chosen cipher– and plaintext attack is the most powerful attack, simply because the attacker has the most information available to deduce the key.

Exercise 2: RSA basics

a) The main arithmetic operation used in RSA is the modular exponentiation.

b) Exponentiation is an expensive operation since there are lots of multiplications (e.g. $(n-1)$ times multiplied by itself) involved. The number of multiplications can significantly be reduced when appropriate algorithms are being used.

Those algorithms are also known under the term "exponentiation by squaring", where the complexity is reduced to at most $log_2 n$ multiplications. Various variants (e.g. left to right binary exponentiation, right to left, k-ary, sliding-window, Montgomery etc.) of those algorithms exist.

A more general method involves addition–chains which may further reduce the number of multiplications. The basic idea behind the algorithm is that the exponentiation can be split up such that intermediate results can be used multiple times (e.g. $a^4 = a^2 \cdot a^2$). Finding a shortest addition chain for a given set of exponents is however a NP–complete problem.

c) The idea of the square and multiply algorithm is to reduce the number of multiplications, that are used to compute y^x, by using the the binary representation of the exponent (e.g. using $y^1 \cdot y^2 \cdot y^4 \cdot y^8 \ldots$ instead of $y \cdot y \cdot y \ldots$) which is more effective. The right to left variant of the algorithm uses two registers to store its intermediate results while the left to right variant needs only one.

Sources: [5, 6, 7]

Exercise 3: Side Channel Attacks

a) A side channel attack is a physical attack that takes advantage of the implementation-specific characteristics. The attacker tries to exploit physical information leakages (e.g. timing information, power consumption, E-M radiation) such that he is able to draw conclusions on the secret parameters that are involved in the computation.

b) A successful side channel attack was the timing attack on RSA. Since the square and multiply algorithm has key–dependent branching it was possible to measure the time needed for the exponentiation algorithm. The multiplication is only performed if the corresponding key bit is set to one which allows the attacker to reconstruct the private key bit by bit.

c) Title: Introduction to Side-Channel Attacks
URL: `http://www.dice.ucl.ac.be/~fstandae/PUBLIS/42.pdf`

d) The maximum attack–distance does depend on various parameters. Using the Biot–Savart, law which explains the magnetic field that an electric current generates, one notes that the current is such a parameter.

$$d\mathbf{B} = \frac{\mu_0}{4\pi} \frac{I d\ell \times \hat{\mathbf{r}}}{r^2}. \tag{1}$$

But the maximum–attack distance does not only depend on the current itself but also on the EM–shielding, the coupling path (e.g. radiative or conductive) as well as the signal to noise ratio.

Sources: [1, 2]

Exercise 4: SPA attacks on RSA

a) As noted earlier the square and multiply algorithm has key–dependent branching (e.g. when the key bit is zero no multiplication is executed) one can use a simple power analysis because the instructions have different power consumption signatures. If the squaring is followed by an other squaring the bit of the exponent is zero. Hence one can "visually" extract the private key.

b) Devices like smart cards that do depend on external power supplies and probably also on the clock signal are most vulnerable to those attacks since they can be monitored and controlled easily from the outside (without leaving any traces).

c) To prevent SPA the execution flow should be as constant as possible. This can either be done using specific hardware that has the same power consumption signature for each instruction or by always calculating the multiplication but then ignoring it when the key bit is zero. Adding randomness to the calculation e.g. by adding a random multiple of $\phi(n)$ to the exponent before each exponentiation could make the analysis more difficult but at the same time might be itself subject to a power analysis.

d) The new design would be less efficient, as the proposed hardware changes would increase the cost significantly and the additional steps that are introduced by the multiplication deteriorate the performance.

References

[1] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-00409-7. doi: 10.1007/3-540-36400-5. URL `http://www.springerlink.com/index/10.1007/3-540-36400-5`.

[2] Prof. Jean-Jacques Quisquater and Dr. François Koeune. Side channel attacks: State-of-the-art, 2002. URL `http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1047_Side_Channel_report.pdf`.

[3] The Internet Society. RFC 2828: Internet Security Glossary, 2000. URL `http://www.ietf.org/rfc/rfc2828.txt`.

[4] William Stallings. *Network Security Essentials: Applications and Standards*. Pearson Prentice Hall, 2007.

[5] Mads B. Tandrup, Martin H. Jensen, Rasmus N. Andersen, and Therese F. Hansen. Fast Exponentiation In practice, 2004.

[6] Wikipedia The Free Encyclopedia. Addition-chain exponentiation, . URL `http://en.wikipedia.org/wiki/Addition-chain_exponentiation`.

[7] Wikipedia The Free Encyclopedia. Binary exponentiation, . URL `http://en.wikipedia.org/wiki/Binary_exponentiation`.