Introduction to Cybersecurity Elisabeth Oswald

Asymmetric Cryptography

2023

Overview

Some exercises are based on paper and pencil. Unless you use a tablet (with a pencil) you must bring actual paper and pencil along.

For some parts of this lab you need to program in Python, and we expect that you have a reasonable level of proficiency in Python already. We also expect that you have Python installed on your system. You will also need the Cryptography library as well as PyCryptoDome.

We do not expect that you can finish all exercises within the 3hr lab session. We also do not require that you do all exercises. The exercises are design to help you finding the answers to the corresponding Moodle quiz.

What you should get out of this.

Today's session's main goal is that you "use" a range of asymmetric primitives and schemes in a somewhat "hands-on-manner" in order to deepen your understanding of them. The emphasis is here on "using schemes" rather than deep mathematical understanding: you are more likely in any job later on to become a "user of cryptography" than a cryptographer. As a user, sometimes this role is also called "Security Archtitect", your job is to design and specify how cryptography is used to achieve certain security goals in a concrete system. Therefore you need to understand what guarantees schemes give you, and you need to be aware that "small/innocent looking" modifications to schemes (standards) can have a devastating impact on security.

Assessment

There is no required submission for this lab, but doing the tasks will enable you to answer questions in a small Moodle quiz which is due at the end of next week.

Paper and Pencil Exercises

Exercise 1: What is a security reduction?

Explain, using your own words, what a security reduction is. What is it used for, and how is it used? Try to explain one reduction that relates to the security of either a DLP or a Factoring related public key cryptosystem from the set of slides. Your explanation should be informal, so you should try and capture the idea/logic. Then, discuss it either with your neighbour (in class) or explain it to your group (online), or talk to one of the teaching staff to see if you got the idea/logic right.

For reductions related to the Factoring problem: find out what is the currently best factoring algorithm? For reductions to the DLP: find out what is the currently best DLP solver? Finally, consider what impact does quantum computing have on the factoring or the DL problem?

Exercise 2: RSA based on a square?

In the RSA cryptosystem we must choose two prime numbers p, q to produce the modulus N. Finding large prime numbers is not a hard problem, however, a friend of yours must implement RSA on a small device where everything is considered "too much". You friend has an idea for speeding up the key generation and asks you for you expert opinion. His idea is as follows: rather than generating two different prime numbers p and q he suggests to just generate a single prime number p and then derive $N = p \cdot p$. Show, either using some informal but somwewhat mathematical reasoning, or via a formal mathematical argument, why this is not a good idea. Provide a concrete example (i.e. with actual numbers) to illustrate your point.

Programming Exercises

Exercise 3: RSA encryption

Using the cryptography library (you must use the hazmat layer) demonstrate encryption with RSA. You should implement RSA encryption in two ways: firstly, implement the encryption of multiple block messages using RSA-OAEP; secondly, implement hybrid encryption using a suitable RSA as a DEM and AES in a suitable mode as a DEM.

Exercise 4: Create a self signed certificate

Based on the provided script `cert_demo.py` generate a self-signed certificate for you as a student of AAU (modify the necessary fields). You certificate should using a DSA or ECDSA type signature.

Exercise 5: Showcase a Man-in-the-Middle Attack

We briefly touched on the idea of Diffie and Hellman for a key agreement protocol. Their original protocol is susceptible against man-in-the-middle attacks. Implement the DH key agreement protocol with realistic parameters using a Crypto library of your choice (in Python). Demonstrate, using your implementation, how such a MITM attack works. Then prevent the attack by introducing signatures in the style of the STS protocol (find out how this looks like).