Project for Cybersecurity

Mandatory requirement to be allowed to register for the examination.

Organization

- Form groups of 4-7 people.
  Need good mix of skills!
- Get organized as group
  Need one group organizer!
- Chose a task.
- Organize the work: Who does what (in writing).
- Server needed: Contact Doritt Linke
- Suggest: Cooperation, social coding: Github.
- Documentation is important.

Requirements

Important:

- Concepts and their description.
- Code and documentation of code.
- Organization of repository.
- Presentations of the groups.

We will have regular presentations on demand and weekly opportunity for discussions.

Goals

Practice some of the tools and protocols discussed in the course.

Train research skills.

Learn cooperation and social coding skills.

Coin Flipping by Mail

Situation:

- Alice and Bob want to do the same task for the cyber security course.
- Carol, their prof, insists that they do different tasks.
- Alice and Bob agree to flip a coin.
- Corona restrictions disallow meeting in person.

Task:

- Provide the tools for a "fair coin flipping over the phone" (or, rather, a "fair coin flipping by mail" protocol).
- Implement them and provide a short description for Alice and Bob.

Constructing General Access Schemes

We know:

- Shamir secret sharing can be used to implement general access schemes.
- We know the mechanism in principle from an example.

Task:

- Implement the mechanism.
- Input: An access scheme.
- Output: The share distribution table.
- Hint: There are many completely different techniques which can be used for an implementation. Java(script), Mathematica, Prolog

Dining Cryptographers

We know:

- The dining cryptographers (DC) protocol allows secure anonymous communication.
- We study the problem for $n = 3$ participants.
- The protocol can be generalized to arbitrary $n \geq 3$.

Task:

- Provide a web-based visualization of the DC protocol for arbitrary $n = 5$ participants.
- Produce examples which demonstrate working scenarios of the protocol.
- Produce examples which demonstrate disruption attacks.
- Use a random generator to control the simulation.
- Goal is a web page which illustrates the protocol for the learner.

Implement Rainbow Table Attack

Task: Launch a rainbow table attack and invert a hash function.

Situation: Given a hash value, find a preimage.

Problem: For "real" hash functions too complicated.

Solution: Use a shortened hash function.

If $f : A^* \to \{0,1\}^{256}$ is a hash function mapping $w \mapsto f_1(w)f_2(w)\dots f_{256}(w)$ then $g\colon A^* \to \{0,1\}^k$ defined by $g(w) := f_1(w)f_2(w)\dots f_k(w)$ is the k-prefix weakened hash function.

Use SHA-3 as f.

Find k such that the k-prefix weakened hash function can be inverted in 30-40 minutes computation time of a standard PC. How big is k?

Feige-Fiat-Shamir Protocol

We learn:

- Feige-Fiat-Shamir (FFS) protocol is a zero-knowledge identification scheme.
- Peggy proves to Victor that she knows the modular square roots of k numbers.
- Victor verifies this claim but does not learn anything about the numbers.

Task:

- Implement a Web-based demonstrator for the FFS protocol.
- Visualises the protocol between Peggy and Victor.

Recommendations:

- Use a node.js / express or a python server.
- Use existing multiprecision arithmetic packages.

Three-prime RSA algorithm

We know:

- The RSA algorithm works with $n = p \cdot q$ a product of two large primes.

Task:

- Generalize the RSA algorithm to $n = p \cdot q \cdot r$, a product of three large primes.
- Formulate the theory of this algorithm.
- Provide an implementation of it.
- Give an example of an encryption and signature using this algorithm.

WAV Audio Steganography

We learn:

- Audio files can hide steganographic messages via LSB and stereo encoding.
- Develop a prototype of a steganographic WAV encoder.

Task:

- Familiarize yourself with the (easy) WAV audio format.
- Construct a steganographic encoder using LSB and stereo encoding.
- Generate samples, encoding increasingly long text in 1-minute audio streams.
- Study the audio degradation with increasing message lengths.

Recommendation:

- There are numerous WAV encoders and audio recorders in the public domain.
- Good examples can be found on npm for node.js.

Anhang

Verzeichnis aller Folien

1. Tasks

Legende:

⧉ Fortsetzungsseite
○ Seite ohne Überschrift
🖼 Bildseite