Fuzzy Fingerprints
Attacking Vulnerabilities in the Human Brain

Konrad Rieck (kr@roqe.org)
On behalf of The Hacker's Choice - http://www.thehackerschoice.com

September 3, 2002

Contents

1 Introduction

Welcome to the world of *Fuzzy Fingerprinting*, a new technique to attack cryptographic key authentication protocols that rely on human verification of key fingerprints. It is important to note that while fuzzy fingerprinting is an attack against a protocol, it is *not* a cryptographic attack and thus does not attack any cryptographic algorithm.

This document covers the theoretical background and the generation of fuzzy fingerprints and also details on the implementation `ffp` [FFP] and its usage. For people who don't want to waste their time reading pseudo-academic Blabla it is essential to skip to the more pratical part of this document 3, the details on the implementation and the provided sample session using SSHarp [SFP].

2 Theoretical background

2.1 Key exchange using public-key cryptography

Asymmetric cryptography has revolutionized the classic cryptography and created new cryptographic techniques such as hybrid cryptosystems or digital signatures. In order to cover the background of fuzzy fingerprinting, this document focuses on the hybrid cryptosystems and their key exchange protocols. Fuzzy fingerprinting may also have an impact on digital signatures or integrity verification systems, for now we simply ignore these aspects.

Let's introduce the classical problem of communicating using a symmetric cypher. Two parties that want to encrypt a communication using a fast symmetric cipher need to exchange a secret session key before starting to communicate. This problem is not easy to solve, meeting in real life or exchanging the session key via telephone are solutions, but often impossible to realize.

Using public-key cryptography both parties can elegantly and securely exchange the session key: Both parties first exchange their public keys. Then one chooses a session key and transmits it to the other encrypting it with its public key. Both continue communicating using the session key. An outside attacker is not able to able to read the secret session key if he just passively eavesdrops the communication of both.

While public-key cryptography looks like a really good solution to the problem, it introduces a new problem into the scenario. An active attacker might intercept the communication between both parties and replaces the transmitted public keys with his own public key. Both parties would exchange keys, but in fact each would receive the public key of the attacker. Any communication first goes to the attacker who decrypts the messages using his private key and then re-encrypts them using the target's public key. He's now able to read the session key in cleartext and can also read the following secure communication that uses this session key. This attack is known as *man-in-the-middle attack*.

2.2 Cryptographic fingerprints for key verification

Several protocols have been proposed to prevent man-in-the-middle attacks when using public-key cryptography, e.g. the interlock protocol [ILP]. Other protocols rely on digital signatures or trusted key distribution centers to verify the integrity of the public keys. Unfortunately in most situation such methods are not available and the initially exchanged public keys are verificated using so called *cryptographic fingerprints*.