STRONGER KEY DERIVATION VIA SEQUENTIAL MEMORY-HARD FUNCTIONS

COLIN PERCIVAL

Abstract. We introduce the concepts of memory-hard algorithms and sequential memory-hard functions, and argue that in order for key derivation functions to be maximally secure against attacks using custom hardware, they should be constructed from sequential memory-hard functions. We present a family of key derivation functions which, under the random oracle model of cryptographic hash functions, are provably sequential memory-hard, and a variation which appears to be marginally stronger at the expense of lacking provable strength. Finally, we provide some estimates of the cost of performing brute force attacks on a variety of password strengths and key derivation functions.

1. Introduction

Password-based key derivation functions are used for two primary purposes: First, to hash passwords so that an attacker who gains access to a password file does not immediately possess the passwords contained therewithin; and second, to generate cryptographic keys to be used for encrypting and/or authenticating data. While these two uses appear to be cryptologically quite different — in the first case, an attacker has the hash of a password and wishes to obtain the password itself, while in the second case, the attacker has data which is encrypted or authenticated with the password hash and wishes to obtain said password hash — they turn out to be effectively equivalent: Since all modern key derivation functions are constructed from hashes against which no non-trivial pre-image attacks are known, attacking the key derivation function directly is infeasible; consequently, the best attack in either case is to iterate through likely passwords and apply the key derivation function to each in turn.

Unfortunately, this form of "brute force" attack is quite liable to succeed. Users often select passwords which have far less entropy than is typically required of cryptographic keys; a recent study found that even for web sites such as `paypal.com`, where — since accounts are often linked to credit cards and bank accounts — one would expect users to make an effort to use strong passwords, the average password has an estimated entropy of 42.02 bits, while only a very small fraction had more than 64 bits of entropy [15]. In order to increase the cost of such brute force attacks, an approach known as "key stretching" or "key strengthening"[1] can be used:

E-mail address: `cperciva@tarsnap.com`.

[1] The phrase "key strengthening" was introduced by Abadi et al. [8] to refer to the process of adding additional entropy to a password in the form of a random suffix and verifying a password by conducting a brute-force search of possible suffixes; but the phrase is now widely used to mean the same thing as "key stretching".