Large Key Sizes and the Security of Password-Based Cryptography

Kent D. Boklan, Queens College, USA

ABSTRACT

We expose a potential vulnerability in the common use of password-based cryptography. When employing a user-chosen password to generate cryptographic keys which themselves are larger than the digest size of the underlying hash function, a part of the resulting key is produced deterministically and this, in turn, may lead to an exploitable weakness. [Article copies are available for purchase from InfoSci-on-Demand.com]

Keywords: AES; Block Ciphers; Encryption; Hash Functions; Keys; Password Based Cryptography; Randomness; Rounds

INTRODUCTION

The genesis of this article came in 2006 when the author was reviewing cryptographic implementations used by certain New York City agencies to protect sensitive user-specific information - in a password-based encryption environment. It was observed that large key variants of well-established block ciphers such as Blowfish and Twofish and 3DES (“Triple DES”) and the Advanced Encryption Standard (AES) Rijndael were regularly being employed. Specifically, 256 bit key Twofish was used (and not the “default” 128-bit key version).

Upon inquiring why these (strong) versions were being used, the answer received was simply that, “larger keys mean[t] greater security”. This note is meant to debunk this myth in the context of password-based cryptography and to suggest that employing “exaggerated” key sizes in this kind of password-based environment may lead to a new type of security vulnerability.

PASSWORD-BASED SECURITY

Password-Based Encryption (PBE) derives an encryption key (see the Appendix for a
Glossary of Technical Terms) from a user-assigned password; it is most commonly used to encrypt files which are stored locally. The Java Cryptography Extension (JCE) provides one framework for PBE. The Bouncy Castle cryptographic API (Bouncy Castle) is another (for both Java and C#). Coupled with a suite of available symmetric encryption schema from which to choose, one has all that is needed to protect data in such a way as that “only” the user in possession of the password can gain access.

Password-based security is invariably less secure than ideal. The primary reason for this is that there is insufficient entropy (i.e. number of bits of randomness) in a password to generate a strong cryptographic key. Keys used in symmetric encryption protocols should be randomly generated; when they are derived from a user password, there’s almost always a marked drop in security. Effective randomness in password selection is very difficult to achieve and still more difficult to demand. (We differentiate between keys for symmetric cryptography and for public key cryptography; the private keys for the latter class are mathematically dependent upon the public keys.) One of the hallmarks of cryptographic best practices is having secrets that are not biased so no information may be gained by an attacker; attacks can be mounted given even slight predisposition. Hidden Markov models are one tool used to exploit such weaknesses and build likelihood distribution models. NIST Special Publication 800-63 (NIST, 2006) suggests than an 8 character, user-chosen password, contains between 18 and 30 bits of entropy. Since an ASCII character can be regarded as a 7 bit number, a truly random 8 character password would have a (maximum) entropy of 56 bits. The average password, so, is not very random, at all. A phishing attack on MySpace reported in late 2006 (Schneier, 2006) gave further insight into password composition for 34,000 users. The good news for security: ‘password’ was no longer the most commonly used password (it ranked 4th). The bad news: it’s ‘password1’ (used in about 1 in ever 450 accounts). The average password length was 8 characters and 1 in 6 used an anemic 6 or fewer characters. MySpace did not place any structural requirements on passwords and it seems that users today have been led to believe that adding a digit to a password enhances security. This is, unfortunately, not a practical truth. Of the MySpace passwords, about 1 in 5 were lower case letters followed by the single digit ‘1’. In all, 8% of users had a password that was a single dictionary word followed by a ‘1’. This sets the stage for a clear attack: assume a password length of 8 characters with a final digit of 1. Now you have only 7 characters to figure out and you’ve removed about 99% of the cases. (When users are required to change their passwords, they cycle the ‘1’ to a ‘2,’ a predictable path of least resistance – so, very often, no security is gained.) Forcing password structure requirements (such as the necessity of containing a number) goes to lessen password entropy and this, in turn, makes data secured by PBE less secure.

A dictionary attack has some chance of success of identifying a password. An exhaustive attack is deterministic but costly. A common measure to “block” dictionary attacks and other password crackers is to salt passwords, to add some randomness – and thereby enlarge the password space. A salt is a randomly generated character string of some fixed length k that is concatenated with the user password before it is hashed and stored. By adding k bits of salt, a dictionary-type attack should take...
Related Content

Open Project Planner
[www.irma-international.org/article/open-project-planner/67376/](www.irma-international.org/article/open-project-planner/67376/)

An Information Security Model for Implementing the New ISO 27001
[www.irma-international.org/chapter/an-information-security-model-for-implementing-the-new-iso-27001/123535/](www.irma-international.org/chapter/an-information-security-model-for-implementing-the-new-iso-27001/123535/)

Privacy and Security in E-Learning
[www.irma-international.org/chapter/privacy-security-learning/23094/](www.irma-international.org/chapter/privacy-security-learning/23094/)

Impact of National Culture on Business Continuity Management System Implementation

Resilience Principles for the ICT Sector
Scott Jackson (2013). *Critical Information Infrastructure Protection and Resilience in the ICT Sector* (pp. 36-49).
[www.irma-international.org/chapter/resilience-principles-ict-sector/74624/](www.irma-international.org/chapter/resilience-principles-ict-sector/74624/)