HashDice – strong, easy to recover passwords

An innovative method for on-demand creation and recovering of strong passwords for on-line services and off-line applications with ability to restore such passwords from a single remembered secret is described. The method allows for change of passwords regularly or on-demand without change of the secret. The method does not require any special software and may be implemented on almost any general-purpose computer.

BACKGROUND

Use of login – password pair is the most widely used method of authentication of users on the Internet. Passwords are also widely used to provide access to off-line resources such as encrypted databases.

Strength of passwords is crucial in many cases, as passwords protect important personal information, trade secrets and access to bank accounts. Passwords should be hard to guess and they also must be long enough to make exhaustive search either impossible or impractical.

Many web services have requirements for format of user's passwords in attempt to improve security of user's accounts. It is often required from the user to compose a password that contains letters in both upper and lower case, numbers and special symbols.

Another important aspect of password security is password uniqueness. It is considered important to have different passwords for different web resources because otherwise stolen from one resource password gives access other accounts of a person.

Each user may have dozens of different accounts. As strong passwords are usually hard to remember and users have to remember many passwords, techniques for securely storing passwords are used.

One known solution for storing passwords is password managers. Password manager is a computer program which keeps user's passwords in an encrypted database. User must enter yet another password, called master password, to get access to the database.

One drawback of password mangers is that the user of a password manager may lose access to its passwords if database of the password manager is damaged, or a device that is used to store the database is lost or damaged.

Risk of loosing of the password manager's database may be reduced by replicating it and keeping copies of the database on different storage devices, but

it creates necessity of keeping the copies synchronized after each change, which is not convenient.

Another inconvenience is that the user have to bring password manager's database with him all the time in case he needs to login to his account from a new computer.

SUMMARY OF THE INVENTION

The present invention addresses the problem of creating strong passwords for online and off-line services, new password for each new service, with ability to change passwords regularly or on-demand.

There is no need to remember passwords created using present invention as a password for any service may be restored or regenerated when it is needed from a single secret passphrase without disclosing the passphrase to the said service. This means that the user needs to remember only one secret passphrase to restore any of his passwords and this can be done without any database.

Absence of a database for passwords excludes possibility of a loss of the passwords due to storage device loss or failure.

Another yet advantage that arises from the absence of passwords database is that the user does not have to worry about security of the said database.

Another yet advantage that arises from the absence of passwords database is that the user can restore its password almost on any electronic computer from the secret passphrase he remembers without need to carry storage device for that purpose.

According to the present invention passwords are generated by applying a special function, called hash function, to a string consisting of at least 2 parts. One part of a said string is a passphrase and the other part is a name of a service for which password is being generated.

Hash function used for password generation is a function that has at least the following properties:
1. it is deterministic so the same message always results in the same hash;
2. output of the hash function is a combination of characters that may be used as a password or output of the hash function may be interpreted or transformed to a said combination of characters;
3. a small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value;
4. pre-image resistance - given a hash function H and a hash value h it should be infeasible to find any message m such that h = H(m). This concept is related to

that of a one-way function. Functions that lack this property are vulnerable to pre-image attacks;
5. for known strings A, B, unknown string X, and known h1 = H (A || X), where || denotes a concatenation operation, it is impossible to calculate h2 = H(B || X).

Possible example of such a hash function may be well know cryptographic hash function SHA-256. It produces a sequence of 32 bytes. These bytes may be written as a string of hexadecimal numerals that may be used as a password.

Said passphrase is a secret combination of symbols created by the user and it may consist, as an example, of several random words or any combination of symbols that must be kept in secret. As it is the only data that the user must remember in order to use present invention for generation and restoration of passwords, this passphrase may be chosen to be complex and difficult to guess.

Said name of a service is a combination of symbols chosen by the user to uniquely identify the service for which password is being created or restored. This name of a service does not have to be an official name of a website or a service for which password is being created or restored, but it preferably is chosen in a way that make it obvious for the user what name he should use for the said service, so that he does not have to remember the name he used last time for the certain service.

Said name of a service is different for different services. Uniqueness of the name for each service ensures that different passwords are created for different services. Properties of the hash function also ensures that first service is not able to calculate user’s password for the other second service from a password for the first service known to it. This also ensures that user’s passwords for one set of services remains safe even if security of the other set of services was compromised.

In other embodiment of the present invention passwords are generated by applying a special function, called hash function, to a string consisting of at least 3 parts. One part of a said string is a passphrase, the other part is a name of a service for which password is being generated, yet another part is a modifier.

Modifier allows for creating of multiple passwords for the same service with the same secret passphrase. This makes it possible to change a password if security of the service was compromised and the user suspects that his previous password for the service may be stolen. Any unique for the particular service combination of symbols may be used as a modifier.

Password creation with use of modifiers is also may be used for regular change of passwords. Any unique for the particular service combination of symbols may be used as a modifier.

One possible example of a modifier for an annual password change may be a numeric entry of the current year. Another example of a modifier for quarterly change of passwords may be a combination of symbols in a format “yyQn”, where symbols “yy” must be substituted with last 2 digits of the current year and symbol “n” must be substituted with a digit representing number of the current quarter of the year.

DETAILED DESCRIPTION OF THE INVENTION

The present invention, in its one broad aspect, provides a method of generation of different passwords for different services using only the name of the service and one secret passphrase.

In one embodiment of the invention password for a service is created by applying a hash function to a string called hashable string. Hashable string is a concatenation of a user’s secret passphrase and a name of said service. Hash function is applied to a hashable string. Resulting hash value is interpreted as a string of hexadecimal numerals in either upper or lower case. Said string of hexadecimal numerals is used as a password for said service.

When user needs to enter his password for said service to identify himself, he do not need to remember his password for the said service. He may instead generate the password again.

Properties of a hash function ensure that his secret passphrase cannot be determined from the generated password for the service. Properties of a hash function also ensure that it is infeasible to determine password for other services generated using different name of a service from the password of given service. This makes it impossible for a person knowing one password of the user to determine other passwords of the user, generated using the present method.

If a hash function used for password generation is susceptible to length extension attack then the secret passphrase is placed at the end of the hashable string to prevent such an attack.

If a hash function used for password generation is not susceptible to length extension attack then order of string elements in the string is not important, as long as it is the same for all services.

In another embodiment of the invention hashable string additionally contains a substring called modifier. Modifier is used to create different passwords for the same service using the same secret passphrase. Modifier may be used to change password for a service if there is a risk that the previous password has become known to other people.

Modifiers must be different for different passwords created for the same service. One example of modifiers may be substrings “1”, “2”, “3” used to create 3 different passwords for the same service.

Modifier may be placed anywhere in the hashable string before secret passphrase.

Another example of a modifier is a substring, containing number representation of the current year. This modifier may be used for annual change of passwords. For example, in 2019 a modifier “2019” is used, while in 2020 a modifier “2020” is used.

In yet another embodiment of the invention the said hash value is further modified to meet requirements of said service.

Some services have requirements for format of passwords.

For example, many services require that a password contains both upper and lower case letters, numbers and special symbols. Hexadecimal representation of a said hash does not satisfy this requirement as it may contain only numbers and letters from A to F in either upper or lower case.

A combination of symbols called addendum is concatenated to the said hash to satisfy requirements of the service.

Addendum contains at least one symbol from each set of symbols that are required to be present in a password.

Addendum is not required to be secret or to be different for each password because it is needed only to satisfy rules of the service.

For example, “!1Qq” is a possible addendum to satisfy requirement for a password to contain special symbols, numbers, upper and lower case letters.

In yet another embodiment of the invention the said addendum contains at least one symbol from each set of symbols that are required to be present in a password, except for symbols that are already present in the hash with high enough probability.

For example, if hash contains 32 hexadecimal numerals with symbols A to F in upper case, and it is known that the numerals are uniformly distributed, then probability that the generated hash does not contain any number is less than 2.4 * 10^(-14), and probability that the generated hash does not contain any upper case letter is less than 2.9 * 10^(-7). For example, addendum “!q” might be chosen in this case.
If a password, generated with addendum that contains at least one symbol from each set of symbols that are required to be present in a password, except for

symbols that are already present in the hash with high enough probability, with some small probability does not satisfy service rules, either addendum is changed for this password to satisfy requirements of the service, or different password is generated using different modifier.

In yet another embodiment of the invention the said hash value with or without addendum is truncated to satisfy maximum password length requirement of the service.

In yet another embodiment of the invention the said hash value is truncated to contain number of symbols, representing hexadecimal numbers, equal to number of bytes of entropy in the said constant secret passphrase multiplied by 2 and increased by up to 4 symbols.

Some methods of secret passphrase generation allow calculation of amount of entropy in the resulting secret passphrase.\

For example, when method called DiceWare is used, and a passphrase is composed by randomly choosing 6 words from a known list of 7776 words, it may be calculated that resulting passphrase contains 78 bit of entropy. Said hash value may be truncated to length of 20 hexadecimal numerals.

Truncation of the generated hash value makes password shorter and easier to enter.

Hash functions may produce collisions. This means that there may be several different string that produce the same hash. Truncation of the hash increases probability of a collision. Length of truncated hash may be increased to up to 4 additional hexadecimal numerals to decrease probability of a collision to an acceptable value.

In yet another embodiment of the invention said hash value is encoded in Base64 instead of being encoded as hexadecimal numerals, also known as Base16.

Use of Base64 encoding makes passwords containing the same amount of entropy shorter and easier to enter, but require use of a special computer program to convert hexadecimal representation of the hash to Base64 as most of programs for hash calculation return calculated hashes as hexadecimal numerals.

The present invention may be implemented as a computer program that let the user choose its addendum and remembers the choice, lets the user to choose length of the truncated hash, lets the user to enter names of services, remembers them and lets the user to choose among already used names of services, lets the user to enter modifier. The program also allows user to enter his secret passphrase, then it performs actions of the described method to calculate a password.

The present invention may be implemented as a special purpose handheld computer that performs actions of the described method.