Modified Key Derivation Algorithm

Sarmad Al-Aloussi

Worcester, MA 01604, Pakistan

Abstract
This paper described a modified password strength key derivation algorithm; it is used the key based random permutation (KBRP) method. The key derivation is a process for generating one or more key for cryptography, when the length of key is N then it can generate N! permutation from single password. This key will use in cryptography process. It can present as character, hexadecimal, or binary codes.

Keywords
Key derivative; KDF; Cryptographic.

1. Introduction

The inability of humans to generate and remember strong secrets makes it difficult for people to manage cryptographic keys. Key exchange is an important problem in practice and several schemes have been designed to solve it since the seminal work of Diffie and Hellman [1]. Recently, different works have been published in order to analyze the security of those schemes in various settings (password, public-key, hybrid setting) and security models. Key derivation refers to the process by which an agreed upon large random number, often named master secret, is used to derive keys to encrypt and authenticate data. A Key derivation function (KDF) is a basic and essential component of cryptographic systems: Its goal is to take some source of initial keying material, usually containing some good amount of randomness, but not distributed uniformly or for which an attacker has some partial knowledge and derive from it one or more cryptographically strong secret keys. The number and lengths of such keys depend on the specific cryptographic algorithms for which the keys are needed. We associate the notion of "cryptographically strong" keys with that of pseudorandom keys, namely, keys that cannot be distinguished by feasible computational means from a random uniform string of the same length. In particular, knowledge of part of the bits, or keys, output by the KDF should not leak information on the other generated bits. Typical examples of initial keying material are a Diffie-Hellman value computed in a key exchange protocol, a bit sequence obtained by a statistical sampler (such as sampling system events or user keystrokes), the output of an imperfect physical random number generator, and more.

The main difficulty in designing a KDF relates to the form of the initial keying material. When this material is given as a uniformly random or pseudo random key K then one can use K to key a pseudorandom function (PRF) to produce additional cryptographic keys. However, when the source keying material is not uniformly random or pseudo random then the KDF needs to first "extract" from this imperfect keying material a first pseudo random key from which further keys can be derived using a PRF. Thus, we identify two logical modules in a KDF: a first module that takes the source keying material and extracts from it a fixed-length pseudo random key K, and a second module that expands the key K into several additional pseudorandom cryptographic keys[h]. The Key Derivation Problem. Diffie-Hellman (DH) based key exchanges establish a secure communication channel between two parties by securely negotiating a large random element in a given cyclic group, called master secret. Then, this secret is used to derive keys for encryption and authentication data. These keys must be bit-strings of some specific length uniformly distributed and used as input parameters to symmetric ciphers (for privacy), message authentication codes (for authentication), and pseudo-random functions (for expansion of a seed into a longer bit-string). However, they cannot be initialized with the simple bit-string encoding of the master secret. Even though this secret is indistinguishable from a random element in the cyclic group under some classical computational assumptions, such as the Decisional Diffie-Hellman assumption (DDH), its encoding is not indistinguishable from a random bit-string with a uniform distribution. The entropy of the bit-string encoded secret is indeed high but not high enough to immediately obtain an almost uniformly distributed random bit-string: pseudo-entropy generators are not pseudo-random generators even when only considering the property of computational indistinguishability [2].

There are many approaches to develop a strong key derivation such as: -

1. Functional based: In this approach (also called password-based) key derivation process is a function of a password. It has many applications in the encryption and message authentication.

2. Biometric Based.
In traditional cryptosystems, user authentication is based on functional key derivation of secret keys, which falls apart if the keys are not kept secret (i.e., shared with non-legitimate users). Further, keys can be forgotten, lost, or stolen. The authentication systems based on

Manuscript received January 5, 2026
Manuscript revised January 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.1.5

physiological and behavioral characteristics of persons (known as biometrics), such as fingerprints, inherently provide solutions to many of these problems and may replace the authentication component of the traditional cryptosystems. Biometric Cryptographic Key Generators, or BKGs, follow a similar design: during an enrollment phase, biometric samples from a user are collected; statistical functions, or features, are applied to the samples; and some representation of the output of these features is stored in a data structure called a biometric template. Later, the same user can present another sample, which is processed with the stored template to reproduce a key [33][34].

3.Voice Based. This approach deals with a technique to generate a repeatable cryptographic key on a Programmable Digital Mobile from a spoken pass phrase. The cryptographic key is derived from merely the pass phrase. The cryptographic key is designed to resist cryptanalysis even by an attacker who captures and reverse-engineers the device on which this key is generated.

Although there are different approaches for key deviation, all of them share some features, however, an algorithm is said to be strong algorithm if it has the following characteristics:
The Characteristics of Strong Key:
The Length of Key must enough, to be difficult from broken.
The Key must include alphanumeric not only numbers or alphabetic.
The run length should be less than 8.(when convert the key derivation to binary sequence).
The entropy should close to 1.
The character must distribute randomly.
The Key should not include any word related to the users(such as his name, birthday ...etc)

2. Algorithm Evaluation:

There are two parameters to evaluate the tong of key are shown below:

2.1. Entropy: is the measure amount of information that is missing before received the transmitted message. It is expressed by discrete set of probabilities pi. Shannon Entropy referred to amount of information in a transmitted message. For example, the DNA strand required to uniquely determine the biological characteristics of a human being contains far more codes than is minimally sufficient to transmit the required genetic information. In information theory entropy is a measure of the average amount of information required to describe the distribution of some random variable of interest (Cover and Thomas1991). More generally, information can be thought of as a measure of reduction in uncertainty given a specific message (Shannon1948, Ayres 1994, p.44).

Shannon Entropy
S = final probability space composed of two disjoint events E1 and E2 with probability p1 = and p2 = 1 – p, respectively. The Shannon entropy is defined a

$$H(S) = H(p1, p2) = – plogp – (1 – p)log(1 – p)$$

No.	Function
1	**Red**
2	**Blue**
3	**Blue**
4	**Red**
5	**Red**
5	**Blue**
6	**Blue**
7	**Red**
8	**Blue**
$E(R,B)=-4/8\log_2(4/8)-4/8\log_2(4/8)=1$	

2.2. Run Length: If a byte occurs at least four consecutive times the number of occurrences is counted. The compressed data contains this byte followed by a special flag, called M-byte, and number of its occurrences. The exclamation mark "!" can be defined as this M-byte. A single occurrence of this exclamation mark is interpreted as M-byte during the decompression; two consecutive exclamation marks are interpreted as an exclamation mark occurring with the data.

3. The proposed Strong Key Generation By SHA-1 Algorithm:

A hash function (in the unrestricted sense) is a function h which has, as a minimum, the following two properties:

3.1. compression — h maps an input x of arbitrary finite bit length, to an output h(x) of fixed bit length n.

3.2. ease of computation—given h and an input x, h(x) is easy to compute.

Standards and Technology (NIST) for certain U.S. federal government applications in Figure 1 and Figure 2..

SHA-1 is:
1. The hash-value is 160 bits, and 32-bit chaining variables are used.
2. The compression function has four rounds.
3. SHA-1 uses four non-zero additive constants.

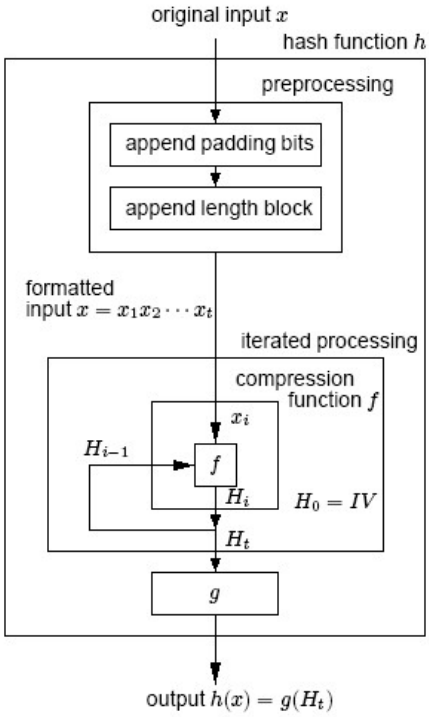

Figure 1. The proposed Algorithm

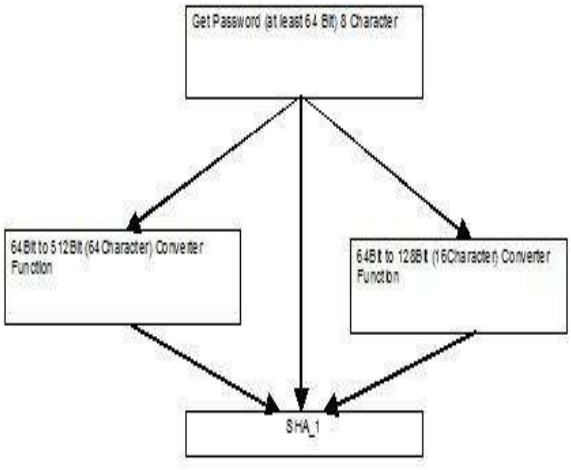

Figure 2. The Block Diagram Algorithm

Steps of execution solving the Conversion Functions.

➢ **Program the following functions:**
- **SHA_1**
- **64Bit to 512Bit Function.**
- **64Bit to 128Bit Function.**

➢ **Execute the Algorithm**

➢ **Test the results by:**
- **Check the Run Length?**
- **Check the Entropy?**

4. Conclusion:

Our work uses a hash function technique to generate strong key. There are many types of hash function it can be used to generate a strong key such these functions is *Collision-resistant hash functions* The most basic security property of a hash function is collision-resistance, which measures the ability of an adversary to find a collision for an instance of a family H. There are different notions of collision resistance, varying in restrictions put on the adversary in its quest for a collision. By applying the Collision-resistant hash functions to produce a strong key and compare the results of new algorithm with the results of the paper under discussion.

In cryptography, CRHF stands for Collision Resistant Hash Function. Also known as Collision Free Hashing Scheme. CRHF consists of a collection of functions. The mathematical concept of a function expresses dependence between two quantities, one of which is given (the independent variable, argument of the function, or its "input") and the other produced (the dependent variable, the value of the function, or "output"). A function associates a single output to each input element drawn from a fixed set, such as the real numbers.

{hs: {0, 1} →0,1}. Such that given s and x it is easy to compute hs(x), but given a random s it is hard to find such that :
hs(x) = hs(x').
This can be characterized by the following two-person game.
Given s (a binary string of length s) and hs().
Find x and x' such that hs(x)= hs(x').
`If x≠ x' then the adversary wins.`

It can modify the same paper by using mod3, mod4 operation at step 4 certify () function and compare the results

References

[1]. W. Diffie and M. E. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, IT–22(6):644–654, November 1976.

[2]. J. H°astad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from any One-Way Function. SIAM Journal of Computing, 28(4):1364–1396, 1999.

[3] Olivier Chevassut1, Pierre-Alain Fouque2, Pierrick Gaudry3, and David Pointcheval2," Key Derivation and Randomness Extraction", eprint.iacr.org/2005/061.pdf.

[4] SOUTAR, C., ROBERGE, D., STOIANOV, A., GILROY, R., AND KUMAR, B. V. Biometric encryption using image processing. In Optical Security and Counterfeit Deterrence Techniques II (1998), vol. 3314, IS&T/SPIE, pp. 178–188.

[5] SOUTAR, C., AND TOMKO, G. J. Secure private key generation using a fingerprint. In Cardtech/Securtech Conference Proceedings (May 1996), pp. 245–252,

[6] Lucas Ballard, "The Practical Subtleties of Biometric Key generation", www.cs.jhu.edu/~lucas/papers/bkg-req.pdf.

[7] Hugo Krawczyk†, "On Extract-then-Expand Key derivation Functions and an HMAC-based KDF ", http://www.ee.technion.ac.il/~hugo/kdf/ for updates.
†IBM T.J. Watson Research Center, Hawthorne, New York. Email: hugo@ee.technion.ac.il

Appendix:

1.Related Functions Definitions:

A hash function usually means a function that compresses, meaning the output are shorter than the input. Often, such a function takes an input of arbitrary or almost arbitrary length to one whose length is a fixed number, like 160 bits. Hash functions are used in many parts of cryptography, and there are many different types of hash functions, with differing security properties There are two functions are specified to produce A key derivation function:

PBKDF1 applies a hash function, which shall be MD2, MD5 or SHA-1, to derive keys. The length of the derived key is bounded by the length of the hash function output, which is 16 octets for MD2 and MD5 and 20 octets for SHA-1.

PBKDF2 applies a pseudorandom function to derive keys. The length of the derived key is essentially unbounded. However, the maximum effective search space for the derived key may be limited by the structure of the underlying pseudorandom function.

2-Related Papers:

"Security Under Key- Dependent Inputs"
"Breaking 104 bit WEP in less than 60 seconds"
"Stateful Public-Key Cryptosystems: How to Encrypt with one 160-bit Exponentiation"
"Distributed Collaborative Key Agreement and Authentication Protocols for dynamic Peer Groups"
"Universally Composable and Forward- Secure RFID Authentication and Authenticated Key Exchange"

3-The Tools for Test the Password:

WLAN strong key generator v2.2 by werewolf Labs.
SecurityStata.com Password Strength Meter www.securitystats.com/tools/
Oracle Password Checker
Pete Finnigan - Oracle and Oracle Security Information www.petefinnigan.com/default/