

# **JUMBLE Public White Paper**



**Pure Entropy Technologies**

## Introduction

Cryptography, the study of hiding information, and cryptanalysis, the study of obtaining hidden information without access to the secret used to obscure the data, are interlinked and complex sciences. The competition between these 2 sciences, akin to the competition between cops and robbers, has been raging since at least the 9<sup>th</sup> century when the first known cryptanalysis document was written or 600 BC when the first ciphers were known to be used. The basic problem is to provide security and usability.

Throughout the years there have been hundreds, if not thousands, of methods proposed to secure information. Most have fallen prey to new techniques in cryptanalysis. DES is an algorithm that was created in the early 1970's and cracked in the 1990's. RC4 is the algorithm that was found to have weak keys with WEP wireless encryption, and because of this WEP wireless encryption is trivial to crack. These algorithms are just some of many that have fallen over the years. Some methods are secure but nearly impossible to implement because of the relative slowness.

Pure Entropy Technologies has created a secure, fast and easily implementable algorithm. JUMBLE is a symmetric cipher algorithm that can operate in block, streaming, marked streaming or hash mode. It is written in ANSI C99 and is compatible with Linux and Windows. The JUMBLE algorithm is currently being implemented on an U.S. government contract to provide security for video and control channels on wireless unmanned ground vehicles (UGV).

## Features

JUMBLE has been designed to operate in 2 native modes. It can be compiled to support either 32 or 64 bit integers. When JUMBLE is compiled in 32 bit mode the key size is 1024 bits, when compiled in 64 bit mode the key size is 2048 bits. For the purposes of this paper JUMBLE compiled in 64 bit mode will be notated as JUMBLE64 and when compiled in 32 bit mode it will be notated as JUMBLE32. If the discussion applies to both compilations of the cipher it will be notated as JUMBLE. JUMBLE32 and JUMBLE64 are not directly compatible. However, JUMBLE32 and JUMBLE64 will run on either 32 or 64 bit platforms. There will be a performance degradation when running a 32 bit compiled version of JUMBLE64 on most 32 bit processors.

JUMBLE64 will accept a key up to 2048 bits. (*Symbol definitions: ^ is "to the power of", x is multiply*) There are over  $3.2 \times 10^{616}$  ( $2^{2048}$ ) possibilities. With JUMBLE64's 2048 bit key it takes  $1.6 \times 10^{616}$  attempts to search  $\frac{1}{2}$  of the key space. JUMBLE32 has a key space of  $1.8 \times 10^{308}$  ( $2^{1024}$ ), an exhaustive search of  $\frac{1}{2}$  of the key space would require  $9 \times 10^{307}$  attempts.

JUMBLE uses no data dependent functions, such as substitution boxes. In a real time streaming environment it is important to operate in a linear time, not a variable time.

Another timing related design is related to parallelization in modern processors. JUMBLE's core engine has no successive serial operations, which allows efficient optimization.

JUMBLE posses an internal tweak value that allows customization of implementations.

JUMBLE has multiple key scheduling algorithms that each possess specific attributes such as being a bijective function, backtracking resistance, cipher block chaining, cipher feedback, and counter based algorithms.

The JUMBLE algorithm has been tested to be resistant to linear analysis, differential analysis, truncated differential analysis, impossible differential attacks, higher order differential analysis, related key attacks, differential-linear attacks, sliding pairs, birthday attacks, side channel attacks, lattice attacks, algebraic definition attacks, statistical attacks and multiple new proprietary and confidential attacks.

JUMBLE is a bijective function that possesses complete avalanche properties.

When JUMBLE is used in stream mode there is no bandwidth addition. When used in block mode there is a slight additive of the difference between the input and the full bit length of the JUMBLE algorithm used. When used in marked stream mode, bandwidth additions are configurable based on the requirements of the particular implementation. A current implementation increases bandwidth requirements are on the order of 1 additional byte needed for 1000 input bytes.

## **Benefits**

Encryption algorithms require speed, ease of implementation, and security.

JUMBLE is a fast algorithm. When JUMBLE32 is run on a 32 bit Pentium IV, this equals about 20 cycles per byte of output. By comparison, the AES256 algorithm takes about 35 cycles per byte of output on the same testing PC, depending on the optimizations used. By using less clock cycles it allows efficient implementation on less powerful processors and requires less power input to run.

JUMBLE is written in C. It can be integrated easily as a standalone executable, Windows DLL, Linux library, or Linux kernel module.

As noted in the features, much cryptanalysis has been done with no weakness exposed. Results and methodologies used are beyond the scope of this paper but are available to interested parties.

