



PET

*Developing innovative encryption
solutions securing the future*



JUMBLE™ Engine Specification Analysis

By Ramie Phillips III

December, 2007

The information contained in this document is proprietary and confidential. This information may not be used by or disclosed to others for any purpose except as authorized in writing by PET. Recipient, by accepting this document, agrees that neither this document nor the information disclosed herein shall be transferred to other documents or used or disclosed to others for manufacturing or other purpose except as specifically authorized in writing by PET. ¹

¹ This White Paper contains information common to both

Table of Contents

1.	Introduction.....	3
1.1	Problem Statement.....	3
1.2	Description.....	3
2.	Market Analysis.....	3
2.1	PET Position.....	3
3.	JUMBLE™ Generator Analysis.....	5
3.1	Cipher.....	5
3.2	Processing.....	5
3.3	Architecture Platform.....	6
4.	Available Solutions.....	Error! Bookmark not defined.
4.1	Competition.....	Error! Bookmark not defined.
5.	Conclusion.....	6
Appendix A:	Glossary of Terms and Definitions.....	7
Appendix B:	Processing Speed comparison to Traditional Encryption Solutions....	10

1. Introduction

1.1 Problem Statement

Too often Data Security enhancement is thought to be a drain on production environments and in many cases has brought production processing to a halt. This paradigm of increasing security and decreasing productivity is the challenge facing Data Security Solutions in the marketplace today.

Pure Entropy Technologies, LLC (PET) and Encryption Security Solutions (ES²) have embraced this challenge by disproving that exact paradigm: “increasing data security, doesn’t have to result in decreased productivity”. The JUMBLE™ engine is a singularly unique, speedy, strong, simple and non reversible encryption solution. JUMBLE’s™ innovative encryption solution yields “increased data security with only nominal production degradation”.

Many business environments lack an end-to-end security strategy due to the large footprint of current security products as well as the inability to secure streaming data. With the development of the JUMBLE™ engine and its non deterministic trap-door function, this technology can be integrated at any architecture tier.

JUMBLE™ offers a robust randomizing capability that applies to many uses and applications within the Data Security market. More importantly, this innovative technology supports identity theft resolution initiatives while requiring virtually no explanation to a customer. Encrypting and de-encrypting is now a simple experience and its use is completely understandable to the observer the moment it is demonstrated.

1.2 Description

PET is the exclusive owner of this unique intellectual property which utilizes up to a 2048 bit key, initially developed for high security governmental applications.

PET has granted an exclusive license to ES² to market and sell a commercial version of the engine which utilizes up to a 256 bit key.

JUMBLE™ is a non-polynomial stream based cipher. This encryption solution is data independent providing rapid processing speed. Since data is independent of the cipher, JUMBLE™ processing handles any form of digital data whether “at rest” or “in motion”. Most notably is JUMBLE’s™ unique ability to encrypt “real time streaming HD (high definition) video.

2. Market Analysis

2.1 Product Differentiation

With the JUMBLE™ portfolio of solutions, increased data security will have virtually no impact on production environments.

Data security customers repeatedly complain that traditional security solutions involve unnecessary complexity, high cost, extra policy requirements, and a degradation of

both business processes and computing environments. In addition, traditional security solutions require extra employee training further impacting production processes.

Applications utilizing the JUMBLE™ engine offer advantages ranging from the simplification of installation, product support, policy requirements and key management.

JUMBLE's™ speedy, strong, simple and non-reversible security solution distinguishes these companies from all others within this Data Security Market. Data throughput using a single core of a Dell Latitude 2 GHz core II duo laptop ranges from 76MB/s to 209.6MB/s.²

JUMBLE's™ design permits completely secure encryption with virtually no impact on bandwidth.

JUMBLE's™ data throughput is up to 300% faster than AES 256, setting this solution apart from all traditional encryption methodologies. (See APPENDIX B)

JUMBLE's™ algorithm can be optimized to prevent compounding traditional latency problems associated with streaming video.

This speed coupled with a 56k uncompressed compiled source code makes JUMBLE™ a singularly unique security solution for any electronic device.

JUMBLE's™ design reduces the traditional drain on battery life.

- JUMBLE's™ small footprint has less code to execute making it quicker while using fewer resources.
- JUMBLE's™ speedy performance decreases energy needed to process the algorithm thereby reducing the power needed for encryption.
- Many security products today rely on toolkit/cryptographic libraries that require more battery life processing procedure calls increasing the number of cache miss. JUMBLE™ is written in C and doesn't use any library calls.
- Top down processing by way of a decision tree provides a faster compute processor than an AES looping techniques resulting in data throughput which is from 12% to 300% faster than AES 256, resulting in a proportionate reduction in power usage, with virtually no impact on bandwidth.

JUMBLE™ has successfully passed simulated testing for use in the low processor, cell phone battery powered, streaming encrypted video environment.

² Throughput is dependent upon the specific algorithm used and the parameter settings.

3. JUMBLE™ Generator Analysis

3.1 Cipher

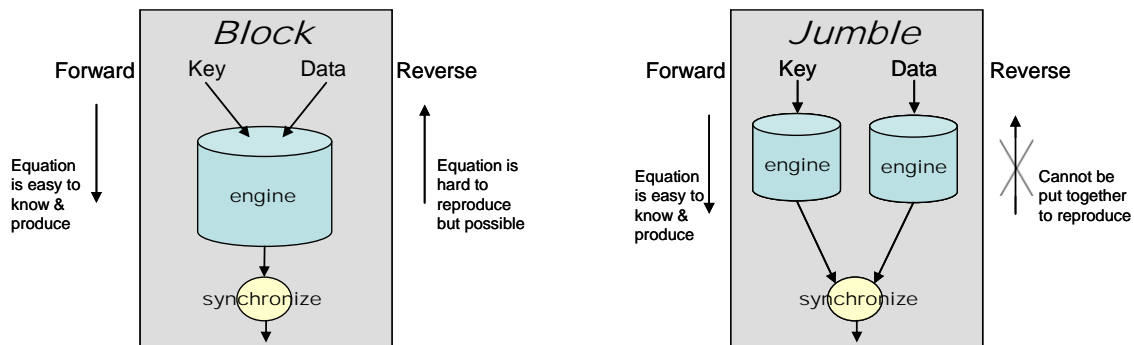
JUMBLE™ is a non-polynomial, non-reversible stream based cipher. Many traditional block and stream ciphers can be reverse engineered back to the source making competitive products breakable, allowing hackers to follow the keys back to the source. JUMBLE's™ innovative design prevents a “backwards” analysis making JUMBLE™ mathematically unbreakable while protecting the source.

When a stream of data enters the JUMBLE™ engine, the XOR'ing factor applied using a derivative of the key, similar to block ciphers producing the output. However, with a block based polynomial cipher, going forward is essentially going backward. In other words, the function of the key and the data together outputs the next function. $f(\text{key} + \text{data}) = f'(\text{output})$. The engine in a block cipher is reliant on the data. JUMBLE's™ non-polynomial stream cipher is a function of the key separate from the function of the data prior to the equation being applied by the engine. $f(\text{key}) + f(\text{data}) = f'(\text{output})$, making JUMBLE's™ engine independent of the data.

It is this separation that allows JUMBLE™ to discard pieces of the key so it cannot ever be synchronized with the data in reverse except by JUMBLE™. Since JUMBLE™ pseudo-randomly discards specific pieces of data, only it can put it back together and synchronize the key to the data at the same time. This separation also accounts for JUMBLE's™ superior high performance.

A block cipher's run is the “invert” i.e., (key + data) or the inverse of what came in. Because the block algorithm is designed to be the invert at some time, eventually forward = backwards. In PET's analysis, since block ciphers are designed to be used in reverse they are theoretically mathematically breakable.

While JUMBLE™ is operating, every state is mixed differently in an ever-changing state with a possibility of 1/256. This is based on discarding residual numbers from the key during the XOR'ing stage.



3.2 Processing

JUMBLE's™ processing methods explain its speed and superior performance. First and foremost there is less data in the engine to process because the key and the data are separate. In addition, JUMBLE™ forgoes the traditional looping method and uses

instead a gate process with decisions made at each gate. In AES 256, the key + data must loop for 14 permutations. With AES 128, the key + data must loop for 10 permutations. Each loop requires computing power and time to complete.

JUMBLE's™ gate process is a top-down approach that provides clear cut on-off decisions.

Both “data in transit” and “data at rest” must be protected against the ever increasing number of sophisticated attacks. Most competitors handle “data at rest” directly with a key + data approach which may leak information about the key and produce patterns that are easily reconstructed. JUMBLE™ starts the engine with a key and only uses the key upon initialization leaving it untraceable back to the source in reverse.

In comparison to other algorithms, this processing method allows JUMBLE™ to use less computing cycles per byte of output.

3.3 Architecture Platform

JUMBLE's™ small code footprint consists of an uncompressed compiled source code measuring only 56k. This allows installation on a wide variety of electronic devices from servers, desktops, laptops, PDA's and cell phones as well as traditional firmware. JUMBLE's™ small footprint provides for implementation at any tier within an architecture platform providing a complete solution for any layer. The JUMBLE™ product suite offers an end-to-end solution by directly installing at the client tier. In some environments, it may be necessary to utilize a network appliance thus a site by site basis of encryption. JUMBLE™, installed on Network appliances can serve several clients and several servers simultaneously.

JUMBLE™ is written in Ansi C 99 language with no standard “C” calls or libraries. This permits JUMBLE™ to run in the kernel as well as environments such as UNIX and Linux. The processing happens inside the windows processor with absolutely no impact on a production infrastructure's data pipeline.

4. Conclusion

JUMBLE™ has conquered the traditional paradigm proving that increased data security doesn't have to result in a loss in productivity. JUMBLE's™ innovative encryption solution provides a faster, simpler and more secure method for protecting both “data at rest” and “data in transit”.

Appendix A: Glossary of Terms and Definitions

Bit	A binary digit having a value of zero or one.
Block	A binary string, for example, a plaintext or a cipher text, is segmented with a given length. Each segment is called a block. Data is processed block by block, from left to right.
Block Cipher Algorithm	A group of functions and their inverses that is parameterized by a key; the function aligns bit strings of a fixed length to bit strings of the same length.
Block cipher	A method of utilizing a block of data to apply bit switches of a key to.
Byte	A group of eight bits that is treated either as a single entity or as an array of eight individual bits.
Data	Inclusive of all forms of digital data including; audio, video, jpeg, images and text.
Decryption	The process of transforming cipher data into plain data.
Encryption	The process of transforming plain data into cipher data.
Forward Cipher	One of the two functions of the cipher algorithm that is determined by the choice of a key.
Inverse Cipher	The cipher algorithm function that is the inverse of the forward cipher function.
Plaintext	Intelligible character data that has meaning and can be read or acted upon without the application of decryption. Also known as cleartext.
Stream Cipher	Continuously applies the bit switches of the key to the data.

APPENDIX B

<u>MB/s</u>	<u>JUMBLE™ with different options 3k startup full load</u>	<u>Difference Mb/s</u>
76.6	Baseline (ci no scrub)	0
88.2	no f 1	92
80.9	no i	34
79.2	no d 3	20.5
76.9	no r 7	2.2
76.8	no c 7	1.4
77.2	no m 3	4.4
75.1	no s 3	-12.3
69.3	add sc	-58.5
59.1	Baseline2 (mi no scrub)	-140.4
	Baseline3 (M no scrub) not	
67.2	ozd	-75.9
61.5	Baseline4 (N no scrub)	-121.3
100.3	Baseline5 (c, c)	189.3
93.8	no f 1, no i	137.6
209.6	Baseline Commercial J16	1063.2

<u>MB/s</u>	<u>Reference Algorithm (true crypt)</u>	<u>JUMBLE™(fastest) speed advantage %</u>	<u>JUMBLE™(full) speed advantage %</u>	<u>JUMBLE™ J16 speed advantage %</u>
56.2	Twofish	78.47%	5.16%	294.48%
52.5	AES 256	91.05%	12.57%	308.19%
51.8	Blowfish	93.63%	14.09%	311.00%
41.3	cast5	142.86%	43.10%	364.65%
32.7	Serpent	206.73%	80.73%	434.25%
12.6	triple des	696.03%	369.05%	967.46%