Nasruddin Cryptographic Function

Ryan Carboni
{ryacko}@gmail.com

This is a 512-bit cryptographic function based upon Rijndael, except the Mix Columns is replaced with Speck.

The Subbytes step could be sped up on computers with AES-NI by using the AES encrypt last round instruction with a subkey of zero and reshuffling the bytes.

The Shift Rows shifts each 64-bit row by n bytes, depending on the position of the row (the top most row is row zero).

Before Shift Rows

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

After Shift Rows

1	2	3	4	5	6	7	8
10	11	12	13	14	15	16	9
19	20	21	22	23	24	17	18
28	29	30	31	32	25	26	27
37	38	39	40	33	34	35	36
46	47	48	41	42	43	44	45
55	57	49	50	51	52	53	54
64	57	58	59	60	61	62	63

The Mix Columns simply operates on each column using the 64-bit Speck round function (word size 32-bits).

Continuing the example in the previous section, the bytes 1, 10, 19, and 28 will be used in the first 32-bit word. The second 32-bit word will be use bytes 37, 46, 55, and 64. The author notes the first part of the Speck round function is essentially a byte permutation, which could be included in Shift Rows in software implementations.

The AddRoundKey simply xors the state with a 512-bit subkey.

Stream Cipher Usage

In stream cipher mode of operation, it functions much like ChaCha. 192-bits of key, followed by 128-bits of nonce, followed by 64-bits of counter, and a 128-bit constant (first 128-bits of pi after the decimal, according to this web page: http://www.befria.nu/elias/pi/binpi.html). It uses no key schedule, and xors the original state into the current state after each round iteration.

While like ChaCha it is vulnerable to slide attacks, the limit to the number of outputs is far below the birthday bound, thus making it an unlikely possibility to find a slide pair, and it still leaks less internal state than RC4, with less biases than both ChaCha and RC4. (The bias as found by Hernandez-Castro, Tapiador, and Quisquater in "On the Salsa20 core function" should not be present in this cipher, as the S-box removes any self similarity between addition and subtraction)

Block Cipher Usage

In block cipher modes of operation, the subkeys are generated through a similar mode as the stream cipher mode of operation.

The first 48 bytes of state uses the key (of whatever length), with zero bytes for padding. The last 128-bits is the aforementioned constant.

Then each key byte is xored with an unsigned byte integer of the round number (with the padding and constant untouched). (making each key length independent from each other)

Then the round function is applied three times, with the original state used for AddRoundKey.

The final state is the subkey for the block cipher round.

The pre-whitening and post-whitening keys is generated by encrypting the integers 0b0, and 0b1, respectively, using the block cipher in ECB mode without whitening. This is essentially using the block cipher in CTR mode. The post-whitening key is added modulo 2^64, while the pre-whitening key is added modulo 2. The author believes that even with independent subkeys, this method of generating whitening keys would make related key attacks vastly more difficult.

It is suggested that 16 rounds be used, except for keys of 192-bits or less, for which 12 rounds are suggested. Obviously no more than 256 rounds could be used.

XXTEA-style mode of operation which expands the block is favored by the author, but to avoid implementation errors, it is suggested that one limits the number of possible block sizes and thoroughly test each one. An XXTEA-style mode of operation is naturally suited to authenticated encryption, if several plaintext words are reserved as a shared authentication secret, as well as a nonce word to ensure uniqueness.

Hash Function Usage

In a hash mode of operation, it uses a variant of Davies-Meyer fast wide pipe construction, except it mixes the entire state more thoroughly.

The cryptographic function is used in a 1024-bit feistel network. The message is prepended with a byte integer of the number of bytes of the hash to be outputted (limited to 64 bytes, in essence the cipher is truncated by at least half). The cipher is designed to be nothing more than 256-bit secure from collisions and 264-bit secure from preimages.

It uses an initialization vector of zero, the internal state is xored with a 1024-bit integer / counter of the number of blocks hashed after each block.

It takes 1536-bits of message. Each 512-bit block is used as a subkey for each of the first three rounds.

The remaining four rounds uses 384-bits from the message and expands it to a 512-bit subkey using the stream cipher mode of operation with five rounds.

In case the block of message is less than 1536-bits in length, zero bytes are used for padding.

Design Rationale

I imagine this is more secure than many other ciphers from Buffer over-read, Cache timing (at least it doesn't use an excess of matrix multiplication), Differential Power, as well as other potential attacks.

A hardware unlimited implementation I believe should not be slower than 64-bit Speck in hardware (the S-box should accelerate diffusion). I speculate that this would improve interoperability between smart cards, and non-portable devices, so that support for Speck wouldn't cripple the cryptographic capabilities of IBM-compatible PCs. It is likely for the indefinite future that hundred dollar CPUs will carry AES-NI support (the US government certainly still uses TripleDES).

The hypothetical hardware unlimited implementation would be capable of pipelining 4 concurrent AES subbytes steps and 8 concurrent 64-bit Speck round encryptions with an iterative Nasruddin design, although obviously it wouldn't be able to encrypt AES or Speck as well as Nasruddin at the same time. This commonality would avoid underutilized chip area for supporting both Speck and AES.

Based on the public literature and rough ballpark estimates, it seems that a hardware unlimited implementation of 192-bit Nasruddin with 12 rounds would have a speed in excess of 100 gigabits per second (given that the fastest AES implementations are 40 gigabits per second), with a security level equal to that of 192-bit AES. This is based on the best cryptanalysis of single-key Threefish being able to attack 12 rounds between key injections, and the author assumes that two rounds of Nasruddin has security equal to that of three Threefish rounds. The whitening keys should provide the equivalent of several additional rounds of security and some provable security based upon the Even-Mansour scheme.

An advantage of the AES S-box, is that despite it's algebraic properties, it is immune to other attacks, and an advantage to ARX functions is that they do not appear to be able to be represented algebraicly. The greatest issue that AES seems to have is optimizing mix columns.

Potential and simple to implement instruction set improvements for the function would be an accessable zero register (already available on many RISC processors), and a solo instruction for AES subbytes (which would speed up Whirlpool, Camellia, and Rijndael-256 (most commonly used for Freenet) as well).

For non-cryptographic purposes, six rounds of Nasruddin is sufficient, as it it should take no more than five rounds to achieve full diffusion. N number of rounds would on average diffuse 3^n number of bytes.

Expected Strength

Nasruddin is expected to have the following security levels for the following parameters:

- 2^{191} applications to bruteforce a 192-bit key with 12 rounds, with a limit to memory or plaintexts of 2^{56}.
- 2^{383} applications to bruteforce a 384-bit key with 16 rounds, with a limit to memory or plaintexts of 2^{80}.

Furthermore, if the subkeys were to be randomly retrieved from memory, it would be impossible to

distinguish the order they are supposed to be arranged in.

Appendix

Patenting: This is not patented to my knowledge, and given it's dependence on Rijndael's general design, and Speck, the author does not believe it is possible to patent this due to prior art, this should not be construed as legal advice.

First 128-bits of PI

The author has a preference for using a non-floating point version of Pi.

0b00100100001111110110101010001000100001011010001100001000110100110001001100011001100010100010111000000011011100000111001101000100

Given that this is from an old web page (http://www.befria.nu/elias/pi/binpi.html) it is unlikely that these digits of Pi were generated in a way to allow for a backdoor.
This should be acceptable given that the Rijndael S-box originates from a table of galois fields.

A Nasruddin joke:

The times were uncertain, and the King's spies read the mail.
Nasruddin did not like that and invented his own language to use with a friend.
An officer of the king knocked on his door the day after he mailed a letter.
"Nasruddin, what is this?" holding up the letter.
"It is in a different language."
"Yes, but which one?"
"I invented it with a friend."
"...what does it say?"
"We disagree on some of the meanings of those words."
"...what about the other words?"
"I forget some of those words. I didn't have much time to learn the language."
"...in the future, I will tear up letters like this."

(there is an infinite number of messages in this joke)