The proof of the existence of perfect secrecy through all-way functions and computational difficulty through multi-way functions in cryptography

Perfect secrecy through all-way functions

The One-time pad offers fast all-way function based encryption with unlimited plausible deniability and guaranteed privacy. All-way functions are those with maximum reversal solutions not exceeding deductive reasoning given the key. Any guess key successfully decrypts all-way function based files and each guess key produces a different output making it impossible to say which output was intended by the key owner for that file. Brute force attacks produce plaintext with every possible symbol combination for that length. The following tables demonstrate all-way nature using modular arithmetic.

Ten-symbol addition possibilities of (a+b) mod 10

There are ten of each digit in the grid. Every column and row contains one of each digit. Every digit yields ten different digits. If only one value is known, the other two are unknown. If two values are known, the third can be logically deduced. There are 10^2 total outputs.

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Three-symbol addition possibilities

plaintext

key symbol	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Similar rules apply for any number of symbols. Here, for every output symbol, three possible input pairs exist and this does not exceed deductive reasoning. For every guess key symbol, only one plaintext symbol is made responsible and for every guess plaintext symbol, only one key symbol is made responsible.

Encryption sample for ten symbols

The output stream borrows random information from the key therefore repetitive data does not degrade file randomness as shown where the plaintext repeats excessively.

random disposable key =	7	2	3	8	7	2	5	3	7	4	8	1	6	9	3	3	2	7	0	9	9	2	4	1	4	1	6	0	2	2
plaintext in base ten =	7	9	2	3	4	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	9	4	5	0	5	3	7	4	9
ciphertext output =	4	1	5	1	1	4	7	5	9	6	0	3	8	1	5	3	2	7	0	9	9	1	8	6	4	6	9	7	4	1

⇧

$(7{+}7) \bmod 10 = 4$

Privacy is guaranteed if random disposable keys are first shared in private. Users planning to share large files must first share significantly large key files in private. Whatever the stored or transmitted encrypted data, its contents cannot be related to you if your copy of the keys is destroyed. And anyone with keys who decrypt your data into something plausible or even illegal still cannot prove the intended output of your data.

Ten-symbol systems are said to have a plausibility ratio of 1:10 and three-symbol systems—a plausibility ratio of 1:3. This ratio may approach infinity and is calculated using the formula: 1 to number of symbols—or to be mathematical—number of symbols to number of symbols squared. Higher ratios compensate for weak random number generation.

The encryption sample for ten symbols demonstrates obfuscation of the fixed-length plaintext of 30 digits. The message itself may be less than or equal to 30 digits since any remaining space is dynamically filled with new random numbers. Had the key length been adjusted to match every plaintext length, the adversary would have a hint at what a short message could be. And had there been no random numbers appended to each message, key digits who drop to fill the remaining output space would give the cryptanalyst a closer look at the exposed key fragments and their generating algorithm.

Computational difficulty through multi-way functions

So far, perfect secrecy has been shown to exist for all-way functions where privately shared keys are used only once and demolished effectively. As the user runs out of keys, such inconvenient protocols require that more keys are shared necessarily in private. Now consider reusing and transforming an existing key or set of keys rather than sharing new ones in private again. This is the immediate step down from the all-way function. Individual output symbols continue with having maximum reversal solutions not exceeding logical deduction, however, with weakness and order introduced to the keys and with more user data to sample—plausible solution possibilities only narrow.

Given only encrypted output as shown in the encryption sample for ten symbols, guessing not only the initial key but the key transformations based on what some algorithm thinks decryption might look like for ongoing messages is a difficult task indeed. Due to the now limited plausible deniability, there still cannot exist reversal shortcuts who compute the intended output for each message therefore search work is inevitable.

With the preferred convenience of one-time privately shared keys, using this proof multi-way function based encryption with key transformation may now be considered a protocol for which decryption difficulty is guaranteed. A key transformed dynamically and symmetrically on both ends in secret can protect large amounts of data before the key transformation details are guessed correctly, if ever. This proof describes a problem classifiable under NP (easily verifiable solutions) but not under P (easily discoverable solutions) therefore P≠NP.

1. <u>All-way functions</u> – Solutions to all-way functions all come to the same conclusion. Each symbol obfuscates a number of solutions equal to the number of different symbols. The number of possible keys to any such symbol or string of symbols is equal to (number of different symbols) raised to the power of (total number of symbols.) Given a disposable key, which solution was intended by the key is the question without answers. Reusing or transforming that key, however, provides a point of entry for the existence of key-data relationships.

2. <u>Multi-way functions</u> – As with all-way functions, solutions to multi-way functions all come to the same conclusion. For example: the hash to which multiple expected strings purposely correspond, where one of those strings may be the intended message or key. And one string solution does not necessarily reveal another. There may or may not be a shortcut to each one of those solutions, however, no shortcut exists for the one intended solution where keys are transformed, and the message is represented using multiple symbols strung together (see Encryption sample for ten digits.) Intended solutions are defined by the generated key and its ongoing transformations. On the other hand one-way functions are not born intertwined and normally do not contain other plausible information on their own.

3. <u>P≠NP is both proven and unproven</u> –
<u>P≠NP is unproven</u> – In one reality, if ignoring key transformation and without cryptanalysis, any values tried for these encrypted file types cannot be mathematically rejected. Conventional verification algorithms not curated for cryptanalysis may consider these trial values to be solutions, however, that is not the stated problem. And it is possible —though difficult—to answer the stated problem…
<u>P≠NP is proven</u> – In the other reality, cryptanalytic search algorithms successfully reject key transformations responsible for random plaintext. Over time and with more user data to sample, the algorithm reveals a transformation responsible for genuine non-random user data, after which any new user messages are successfully decrypted. Such an algorithm treats the multi-way function as a one-way function based on its search priorities. And no reversal shortcuts exist as only through heavy search work it is possible to determine the corresponding shared secret.

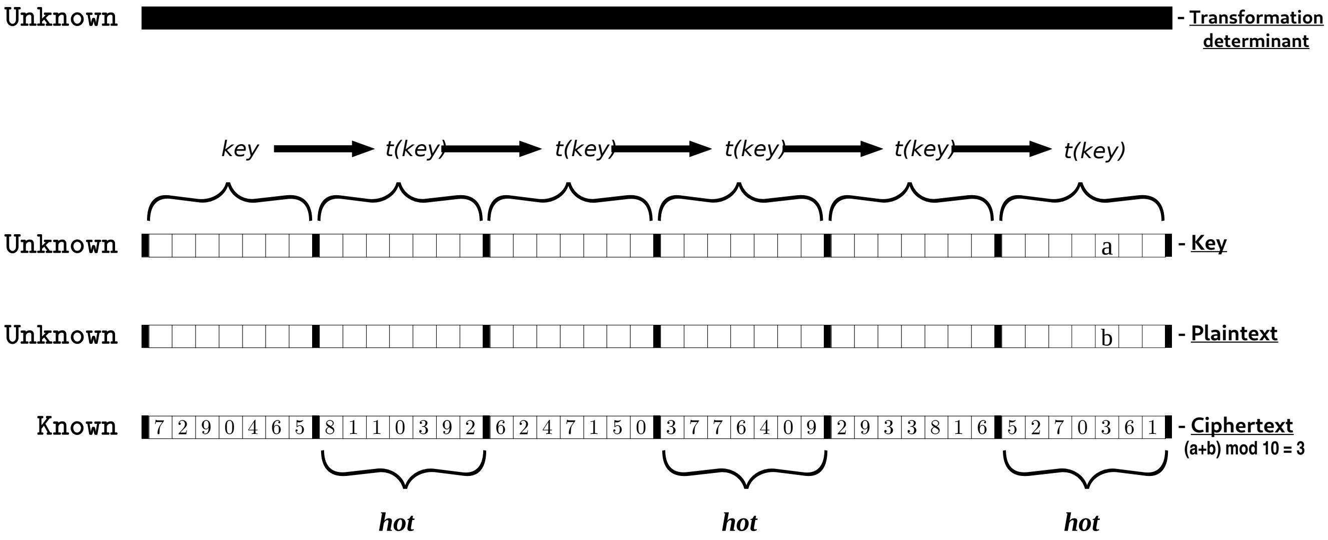
4. <u>The cryptography community is satisfied either way</u> – P versus NP is only a question inquiring about—among other things—information security, the possibilities and impossibilities of crunching large numbers, and whether slightly longer ciphertext poses exponentially greater difficulties. Whether or not you choose that P≠NP has been proven, cryptographers can finally rest assured that perfect secrecy and computational difficulty is possible and here to stay!

5. <u>One-way functions are probably out of the question</u> – Decryption difficulty has been proven to exist by first proving the existence of perfect secrecy, then taking steps down from there. This is comparable to becoming aware of the farthest point of attack, then navigating away and arriving at the solution while having gone through an impractical number of steps. And you could not have determined the usefulness of one step without first having tested the step before that and so on.

Consider looking for shortcuts to some one-way function given correct solutions. Take for example at least two unique semiprimes of cryptographic strength. The objective is to build a single engine who can transform necessarily both semiprimes into their constituent primes, one semiprime at a time, and without using division. This can be tried in reverse where the objective is to transform the four primes into their products without using multiplication. This algorithm's reverse reformulation would then factor semiprimes quickly. Any successful engine is then fed a significant quantity of semiprimes for those who need some assurance of functionality. And there may be multiple engines who do the same thing, however, their function cannot be classified under P or NP unless the proof of reversal shortcut functionality is constructed—at least for some interval, set, or special cases.

It is possible to begin with solutions when looking for shortcuts to those solutions, but where are the boundaries of useful information and is there anything of interest in between? Without an awareness of such a landscape it seems the search is a computational difficulty of its own. See #2 (multi-way functions) for the reasoning behind the exclusion of multi-way functions in this conjecture. Without key transformation, even multi-way functions can achieve perfect secrecy by assigning each key to some complete list of symbols where in comparison to the function's decrypted output, the intended key weight represents the order of its assigned symbol.

6. <u>Computational difficulty in multi-way functions curated for publicly-verifiable authorized-only systems</u> – This point should help visualize computational difficulty and build confidence in raw or layered multi-way functions in CSC (Cryptographic Semiprime Coupling) and its superior with computational difficulty—Authorship. The objective is to construct concise multi-way functions to which one intended key and many-to-none unintentional plausible keys correspond—where the discovery difficulty of any key is strong and roughly equal. Although a single key is preferable, cryptanalytic search priorities for plaintext may successfully verify unintended keys due to the nature of limited plausible deniability as solution possibilities narrow with key reuse or transformation. The following sample multi-way function significantly reduces the verifiable key quantity while retaining the impossibility of any reversal shortcuts based on some built-in search priorities meant for public distribution.



<u>Transformation determinant</u>: only the input values are random. They are the transformation guide in some built-in protocol. **<u>Key</u>:** only the first sub-key is random. Its transformations are predeterminate from there. **<u>Plaintext</u>:** strings can be identified and verified for their symbolic or numerical properties. **<u>Ciphertext</u>:** multi-way function with sub-key to sub-function order correspondence.

Cool sub-functions obfuscate random plaintext. Hot sub-functions obfuscate plaintext with identifying properties expected by the search priorities. For example, each sub-plaintext in a hot zone must be composed of contiguous primes of any length. Here, public decryption may include a string of digits each representing the prime digit length and in order—so as to avoid tree search, ambiguity, and the identification of prime sub-strings within the primes. Zone parameters and the transformation type would be built-in and distributed—along with search priorities.

Whatever the identifying plaintext properties required to pass verification, neighboring sub-functions in the cool zones show unlimited plausible deniability through random output. However, the presence of complete data samples, general search priorities, and predeterminate key transformation establishes the strong relationship between the transformation determinant, key, and plaintext—in congruity with ciphertext open to observation.

Now the function dramatically steps down to multi-way whose cryptanalysis necessarily begins with trial strings in order to determine their transformed usefulness in the neighboring sub-functions hence the persistent difficulty in discovering intended solutions and plausible additional solutions if any—based on corresponding search priority.

Page 1 constructed on February 21 2020
Nikolay Valentinovich Repnitskiy