The proof of the difficulty of disqualifying multi-hasp from P in cryptography

Scope:

Multihasp is a mathematical lock-box with multiple hasps for multiple padlocks. The objective is that Alice and Bob agree upon a secret key while communicating over public networks (homomorphism.) You can either classify functions under P or show that they are not classifiable under P. This proof shows that it is difficult to disqualify Multihasp functions from P; Multihasp security is difficult to guarantee.

Stepping stone:

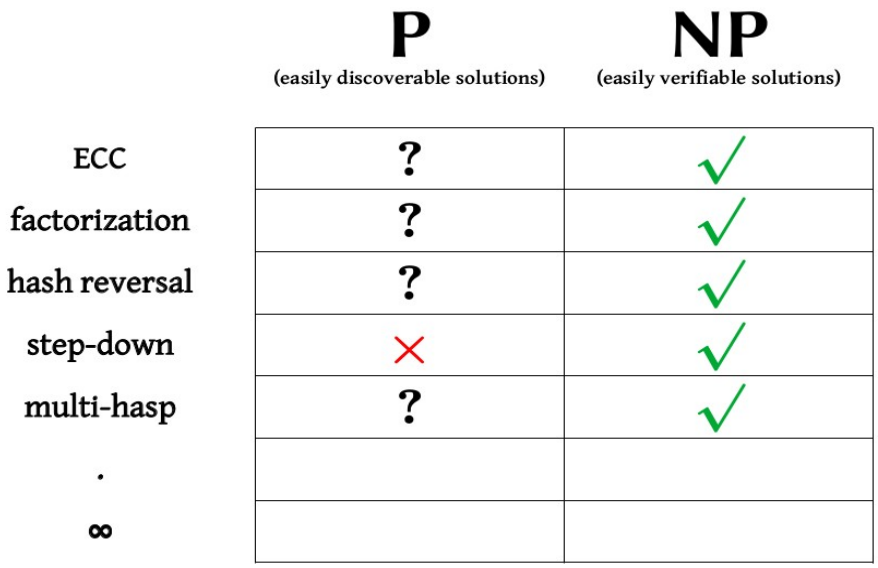
The proof of the existence of perfect secrecy through all-way functions
and computational difficulty through multi-way functions
In cryptography

Assuming conjecture:

#5 in stepping stone above.

Proof:

All multi-hasp systems rely on numerical correspondence between items generated by Alice & Bob. So far, all multi-hasp systems require solutions that are mathematically inherent to their functions. Now multi-hasp systems are equal to one-way functions. On the other hand, solutions to multi-way functions are not mathematically inherent. Multi-way functions (of type step-down) have no reversal shortcuts while one-way functions are out of the question therefore all multi-hasp systems are out of the question.



	P (easily discoverable solutions)	NP (easily verifiable solutions)
ECC	?	✓
factorization	?	✓
hash reversal	?	✓
step-down	✗	✓
multi-hasp	?	✓
.		
∞		

General multi-hasp is listed for side-by-side comparison of classification under classes P and NP. The proof states that it is difficult to put an x under P for all multi-hasp systems.

1. What is Multihasp? The naming convention comes from the powerful idea of a lock-box having multiple hasps for multiple padlocks. You can leave the box out in public where multiple friends apply personal padlocks leaving the box always secured, yet you may remove your own lock to give others access. What does this mean if you have goods or sensitive data in the box? You have transferred goods to your friends—goods handled by any shipment network, yet goods consumed only by those intended, all without anyone ever sharing any keys. Now that's convenience at a distance.

Multihasp is a personal project intended to compete with the only good publicly-known dual hasp system (RSA) known as asymmetric public-key cryptography which utilizes semiprimes, powers, and modular arithmetic. The problem is, there's no mathematical proof that semiprime factorization is free from reversal shortcuts. They are one-way functions and now so must be Multihasp. This equality is concerning because the stepping stone gave us decryption difficulty unrelated to one-way functions—not a single one!

2. Proof review: First, you were asked to conclude that all multi-hasp systems work by having something to correspond with—you cannot agree upon a secret by sharing things that don't have inherent arithmetic properties. Operations can't leverage randomness for conclusion, especially if you must find that randomness through brute force as you would in step-down functions, no matter the key transformation or search priorities for uninherent identifying properties.

Next, you saw that because of this, multi-hasp systems are equal to one-way functions. And if one-way functions are out of the question (pertaining to yes-or-no classification under P) then so are out of the question all multi-hasp systems.

Recall that in step-down functions, you begin with perfect secrecy of the One-time pad then take steps down from there. So you are first made aware of the strongest boundary then the landscape between it and how far you've decided to step down from that boundary. See, boundaries are useful in proofs but Multihasp must have nothing to do with perfect secrecy in order to have leverage for conclusion. So then where are the boundaries of arithmetic tricks for one-way functions? Surely, you must begin with all possible tricks then step down to where one trick unlocks the function (ciphertext) using a default decryption method as distributed among users. But not without also describing the landscape between at least those two points. (Description means real arithmetic key transformation like in step-down. Without arithmetic navigation, how would you transform shortcuts combinatorically and hop between solutions that are shortcuts?)

3. Function piles & tumbler fragmentation: To those satisfied with statistical evidence, this proof is helpful because given new absurd perspectives, it is still difficult to say whether interesting one-way functions have reversal shortcuts. And to those satisfied only with proof and equations for all phenomena, you are closer to eternal dependency on having to share keys privately in person (see #4.) But this only unveils more problems of interest. Consider the following Multihasp which needs only 2 send commands:

1. Alice privately generates a disposable pile of ciphertext and sends it to Bob.

2. Bob privately orders the ciphertext into a list and sends only the list hash to Alice.

3. Alice privately discovers the list order using the ciphertext keys. This order is the shared secret.

The longer the list, the more permutations, which means longer keys (stronger shared secret.) Here, minimalism had reduced the problem to a special kind of hash which reveals the order of what it hashed but only if you own the keys to what it hashed. What's worse, the ciphertext listed might have to be arithmetically related to this hash. (Authorship on the other hand does not depend on ciphertext particularities, it assembles any items of difficulty into ordered lists (see Authorship #16.))

Here's the good part: you need never strengthen or compromise design for authentication because you have Authorship—it takes care of that symbol-for-symbol. So you can carelessly play with Multihasp and include other components if you wish. For example, a tumbler might be constructed beforehand using components from both Alice & Bob. Both can "see" through it so Alice dumps ciphertext into that tumbler which Bob then "shakes" to order the ciphertext into a list. Even better, what if Bob's tumbler components already contain Alice's public ciphertext? Now Alice's components need only be a looking-glass to see the order.

4. Physical Multihasp & microscope-images: Physical Multihasp can also be reduced to other forms of problems. You can reduce the physical security problem to physical tamper-evident devices. Imagine sending keys on SD cards in mailing envelopes. Instead of securing the envelope in a pick-able lock-box, you wrap the SD card with paper, glue the seams, and take microscope-images. The receiver then also takes microscope-images and sends them to you for comparison, while authenticating each image hash using Authorship.

Anyone tampering with the package cannot possibly clone or put it back the way it was within a reasonable amount of time. Now all you need is something more tamper-evident and self-destructive than paper for when, those interjecting mail try poking through the glued paper using micro-conductors to read the chip.

Only when the chip reaches the intended destination can the receiver tear it open carelessly, just after they send you Authorship-authenticated microscope-images of the tamper-evident enclosure of that chip. Now, let's say someone had interjected the package, read the chip, repackaged it, and sent it out to continue. Any microscope-images taken by the final receiver and sent to the sender will mismatch what the sender imaged. So the sender instructs the receiver to discard the package… Keys are no longer strings of characters but recreation of physical material—recreation with numerous specifications.