Sending files over the internet, problems with hashing and thinking ideas.

Problem: Sending a hash over an insecure network instead of an actual file has many benefits, but you don’t get the file, as it is non-reproducible.

Hypothesis: Sending a file bit by bit and letting the receiver reconstruct the file after receiving all the bits. If an attacker manages to retrieve a bit or two, that information would be useless to them. For simplification and understanding purposes, say we split up the sentence “What are you doing?” into four separate entities. “What”, “are”, “you”, and “doing?” Are all sent in packets separately, at randomly timed intervals among placebos that seem legit. The receiver would need some sort of private key that discerns the placebos from the real message. Also, either the real words would need to keep order, or they could have some sort of pointer attached to them that designates their order.

Understanding concepts situation:

* Can 1 bit be encrypted?
* Spaces in a text document are represented by bits, right?

Example:

Message:

“What are you doing?”

Message split up into bits (with spaces):

“What ”, “are ”, “you ”, “doing?”

Pointers added for order:

1 -> “What ”, 2 -> “are ”, 3 -> “you ”, 4 -> “doing?”

Placebos (ended with a #) added (with pointers):

4 -> “doing?”, 2 -> “the #”, 2 -> ”are ”, 4 -> “by #”, 3 -> ”you ”, 1 -> “What ”

Table for clarity:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Label1 | Packet2 | Packet3 | Packet4 | Packet5 | Packet6 |
| Pointers -> | 1 | 2 | 2 | 3 | 4 | 4 |
| Data | What | are | the # | you | by # | doing? |
| Placebo | (no) | (no) | # (yes) | (no) | # (yes) | (no) |
| Timestamp | 0:35 | 1:01 | 0:03 | 1:22.12 | 1.40.00 | 1.59.44 |

Eeeee

Note: The order sent and the timing would be completely randomized within a given constructed time frame, say, 2 minutes. Perhaps at this point, going through this amount of effort, time, and computational resources a company should consider: Is it completely necessary to send this information over the Internet?

Note: It would be nice to send non-placebo, non-related information to the same receiver. The attacker could take all the information in, but not have any idea which set of bits or combination of information deems sensical (O.K. Word, sensible).

As this is just an example, the reality of this concept could get very complicated. All that is needed is a simple program that sends data at random with, contradictory to the example, mass amounts of placebos interlaced in the message. The program needs only to weed the placebos out using its private key. # is an example of that private key.

Complications:

* The computer that uses this program would have to be not connected to the internet because of its availability to the private key, but this information would be received from the internet, so this would add some administrative effort.
* If the attackers obtains EVERY packet, and can completely distinguish the placebos from the real information, and they know the encryption scheme used on each packet, they could possibly decrypt and reconstruct the data, which our program on our disconnected and high computing power computer does (high computing only for decryption).