- There's some differences w/ RDF and RDFS entailment that we should examine.
- Note that blank nodes are local, have no URI, and cannot be exported as an answer.
- This is known as scoping, and scoping is kept to the graph of the Query.
- Practically, the answers are computed as if blank nodes are new nodes w/ URI, and those are only specific to the answer.

Security (Presentations 16 october)
23 october
30 october

- 20 minute presentation
(3ppl → 30 min max)

*Additional details → moodle.

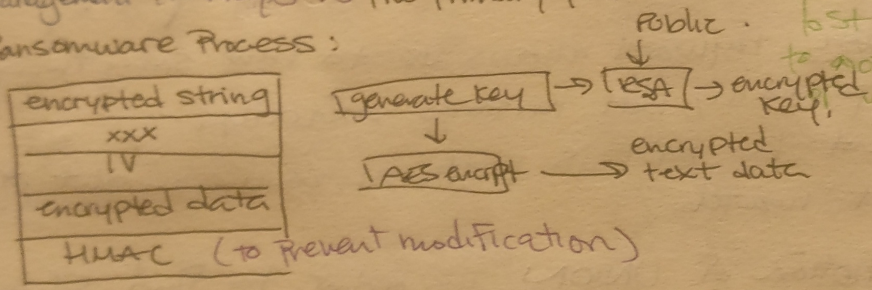
*contact w/ topic, group, and prefered pres day.

maybe we can research topic of that article about that guy who lost a lot of money to not having 2 Fact Auth → whatever it was.

O find it, but maybe we can look into password managers? (entropy, bits, etc??)

Or maybe guessing D/F. ie: Random Number Generators.

- management of Keys is the primary problem.
- Ransomware Process:



encrypted string
xxx
IV
encrypted data
HMAC (to prevent modification)

- Usually the Key is different for each file (aka different encryption for each file targetted.)
- Very rarely target ~~workable~~ files necessary for system, because they want you to pay.
- Usually the Key is generated on your machine.

Kerckhoff's Principle → ① system must be Practically/mathematically indecipherable.
② Should not require the system to be secret.

- The idea that the System should be not obscured. That it does not provide Security.

- Unconditional security: no computing Power can crack an encryption.
- You would require a key as long as the plain-text. (not feasible.)
- Ciphertext should not give you knowledge about the plaintext.

- The "one-time pad" is the principal example of unconditional security.
- More Practical :
  - ① Crypto algorithms that relate to mathematical complex concepts. (ie: Factoring $n \rightarrow p.q$.) "Proven Security" (equivalent, as hard as a mathematical complex problem.)
  - ② computationally good enough (ie: AES, DES, ...) we cannot directly relate to a mathematical problem (like RSA ⟷ Factoring.)
  - These two categories are different, ie: quantum might be able to break RSA but ∅ DES.

- ENTROPY.
  - quality of information, rather than quantity of info.
  - This concept applies to plain-text, cipher-text, and the key. You have to have good entropy for all 3 parts to guarantee good security.
  - High entropy -> increasing randomness.
  - Redundancy is the complement to entropy: we want less redundancy.
  - For instance : ASCII has 8 bits per character, and only 1.3 bits of entropy.
  - observing the cipher-text should not give you info on plain-text.

- Notes regarding Security TP1.
  - note that in multialphabetic substitution, if your key gets longer, it actually becomes harder to Ⓢ. If you have a key that is as long as the plaintext, it becomes a one-time pad. Impossible to crack. (You mistaken it for mono sub because you thought it was replacement, but it's not, it's adding the value, meaning if there's no roll over, you just add "random" values to each character, effectively scrambling it.)

- Even w/ Huffman encoding, there is still a limit to it. It cannot be smaller than the entropy of the alphabet.
- Huffman coding is optimal (mathematically.)