Current and Future Directions.
Computing over encrypted data: homomorphic encryption and CryptDB

Homomorphic encryption
• Encryption Enc is called homomorphic with respect to an operation \( * \) if
  \[ \text{Enc}(x * y) = \text{Enc}(x) * \text{Enc}(y). \]
• That is given encrypted forms of \( x \) and \( y \), in order to compute encrypted form of \( x * y \) one does not need to decrypt \( \text{Enc}(x) \) and \( \text{Enc}(y) \)
• Computations over encrypted values!

Partial vs Fully homomorphic schemes
• Partially homomorphic encryption: with respect just to one operation;
  • RSA (unpadded) is homomorphic with respect to multiplication. Why?
Fully homomorphic schemes:
  • with respect to multiplication and addition
  • Allow to perform arbitrary computations
  • Existence is by no means obvious

Breakthrough
• Craig Gentry: first fully homomorphic encryption scheme is announced by IBM on June 25, 2009.
• The scheme is impractical for many applications:
  • ciphertext size and computation time increase sharply as one increases the security level. Key’s size is also an issue.
Recent developments

- New more efficient schemes and implementations 2010-2016, key size is reduced at least to 600Kb
- **HELib** is an open source implementation (2013)
- More implementations available, including in R and Python
- Still more work is needed to make it practical

Potential applications

- Computations on not entirely trusted services (in the cloud):
  - Encrypt your computational task and send it to a remote server;
  - The server computes over encrypted task/data and returns an encrypted result;
  - Decrypt result;
  - Pipeline processing without revealing intermediate data;
  - ...

CryptDB

Similar idea in data processing:

- To query encrypted SQL database without decrypting;
- Selected fields can be encrypted;
- Practical working prototype system: CryptDB, Raluca Ada Popa et al, MIT (2011-...):
  - http://css.csail.mit.edu/cryptdb/
- Low overhead: reducing throughput 15-25%

Possibilities for MSc Projects

- New applications of existing fully and partially homomorphic schemes;
- Experimenting with and improving CryptDB ideas
- Etc