Implementing a Simple Homomorphic Encryption Scheme

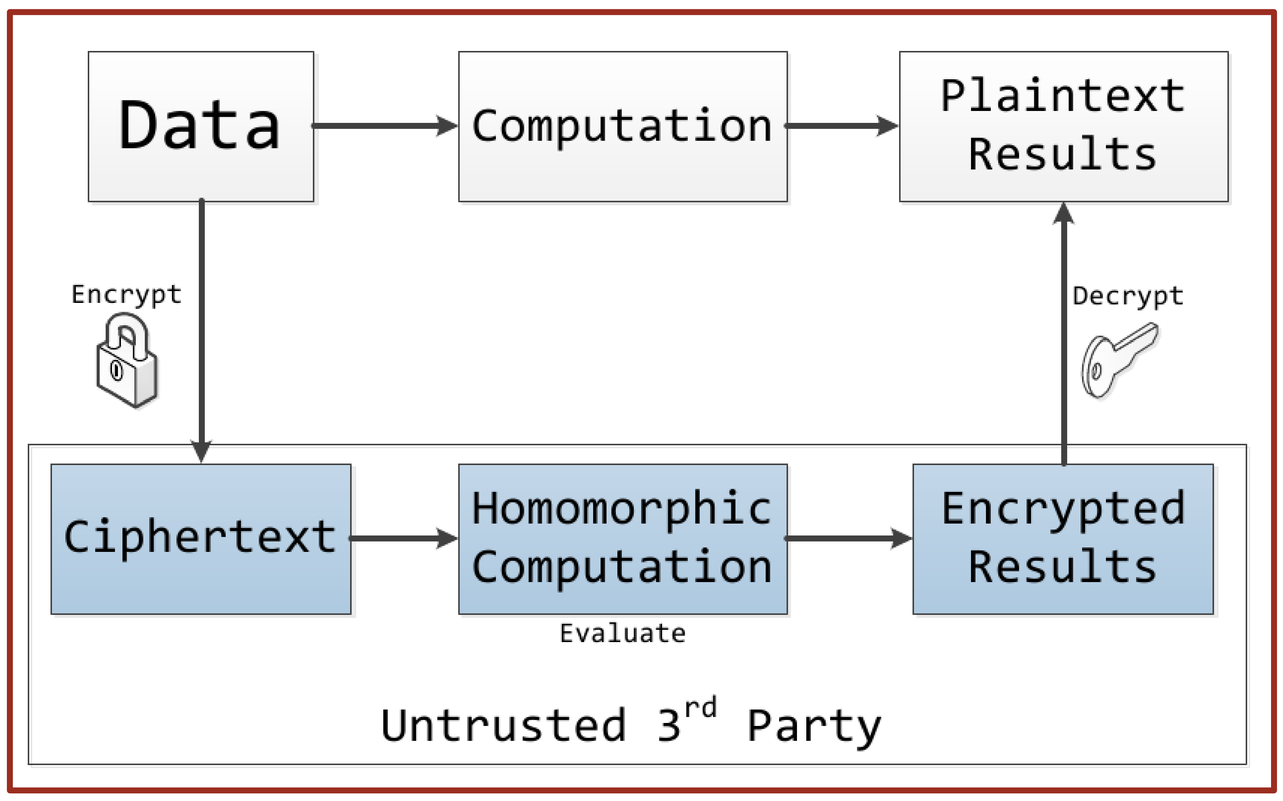
Stephen Crane - Cal Poly Pomona

What is Homorphic Encryption?

Homomorphic encryption allows computation to be performed on encrypted data, without access to the plaintext data or results.

Previous Work

- First Proposed by Rivest, Adleman, and Dertouzo in 1978. [3]
- Proven possible by Craig Gentry in his 2009 PhD Thesis. [1]
- Further improved by Smart and Vercauteren later in 2009. [4]
- Finally feasibly implemented by Gentry and Halevi in 2011. [2]



Encryption Scheme Details

- Gentry's original scheme, and most improved variants of it, use ideal lattices in the encryption process.
- Our implementation uses a scheme published by van Dijk, et. al. [5] which uses only simple integer operations.
- Each bit of ciphertext is a very large integer, along with a large vector of smaller integers.

Homomorphic Computation Details

- Computation is represented by binary circuits of AND and XOR gates, which is theoretically sufficient for any computation.
- These AND and XOR gates are evaluated by simply multiplying or adding the ciphertexts, respectively.
- Each operation adds an amount of "noise" to the ciphertext, and ciphertexts cannot be decrypted after the "noise" has reached a given limit. Thus, the length of a sequence of gates is limited.
- In order to support any possible computation length, this scheme uses Gentry's Recryption algorithm to refresh a ciphertext to remove noise.

Implementation

- We implemented this integer based scheme in C++ using the GMP library for large integer operations.
- Our implementation fully supports encryption, decryption, and evaluation of small circuits
- Unfortunately this integer scheme is not fast enough to be able to practically support the ciphertext recryption needed to support arbitrarily large circuits.
- To support recryption, this scheme would require key and ciphertext sizes several orders of magnitude larger than what is reasonably possible.

Conclusion

- While homomorphic encryption is not useful in a practical sense yet, it still has great promise.
- With the current rate of research and creation of new homomorphic schemes, this revolutionary idea will hopefully one day become a practical reality.
- We hope that this implementation can further this research by demonstrating the concepts of homomorphic encryption in a simple and clear manner.

References: [1] Craig Gentry, A fully homomorphic encryption scheme, Ph.D. thesis, Stanford University, 2009, crypto.stanford.edu/craig. [2] Craig Gentry and Shai Halevi, Implementing Gentry's Fully-Homomorphic Encryption Scheme, 2010. [3] R. Rivest, L. Adleman, and M. Dertouzos, On data banks and privacy homomorphisms, Foundations of Secure Computation, Academic Press, 1978, pp. 169–177. [4] N. P. Smart and F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes., Public Key Cryptography'10, 2010, pp. 420–443. [5] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan, Fully homomorphic encryption over the integers, EUROCRYPT (Henri Gilbert, ed.), Lecture Notes in Computer Science, vol. 6110, Springer, 2010, pp. 24–43.

Author: **Stephen Crane**
Faculty Mentor: **Dr. Craig Rich**
Kellogg Honors College Capstone 2011