Literature Review:
Cryptographic Methods in Machine Learning

Brandon Hosley

July 19, 2020

1 Introduction

The public discovery of asymmetric encryption by Diffie and Hellman in 1976 [1] laid the foundation for an entirely new segment of cryptography and enabled many types of transactions that may have been otherwise impossible. Perhaps the most well-known application of this discovery is public key cryptography which uses properties of asymmetric algorithms to provide digital signature authentication, non-reputability, and the ability to generate shared secrets over public communications.

Common to the most widely adopted asymmetric encryption algorithms is the homomorphic property. While this property garners some attention from the early pioneers of these algorithms, it would take some time before available hardware would catch up to the computational needs. However, this property would prove to be the foundation for much research on analysis of encrypted data.

In the following sections we will review advances, major and recent, in homomorphic cryptography and applications to data analytics with emphasis placed on providing secrecy and security to the machine learning family of algorithms.

2 Important Ideas

2.1 Homomorphic Cryptosystems

Among the most prolific implementations of asymmetric encryption has been the RSA algorithm invented by Rivest, Shamir, and Adleman.[2] RSA expanded upon the principles of the Diffie-Hellman key exchange. Where Diffie-Hellman generates a shared secret from a prime modulus of the product of a series of integers RSA instead publishes the product of two primes and generates two keys based on the primes, publishing one and keeping the other secret.

Keys :

$n = pq$ composite of two large primes

$\langle sk, m \rangle$ Secret key and plaintext

$\langle pk, c \rangle$ Public Key and Ciphertext

Encryption :

$$m^{pk} \equiv c \mod n$$

Decryption :

$$c^{sk} \equiv (m^{pk})^{sk} \equiv m \mod n$$

Figure 1: Generalized RSA Operation as described in [2].

Within the same year Rivest and Adleman, this time joined by Dertouzos published another paper in which they detail a use of RSA as a "privacy homomorphism." [3] This would later be more specifically described as a multiplicatively homomorphic encryption scheme.

Inspired by Benaloh's scheme based on prime residuosity [4], Pascal Paillier conceived a cryptographic scheme based on composite residuosity [5]. He uses this same method to develop a new trapdoor function and a faster decrypting variant.

Encryption :

plaintext $m < n$

select a random $r < n$

ciphertext $c = g^m \cdot r^n \mod n^2$

Decryption :

ciphertext $c < n^2$

plaintext $m = \dfrac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n$

Figure 2: Paillier's composite residuosity based scheme. [5, p. 7]

In the years that followed numerous other asymmetric cryptosystems were introduced that were similarly partially homomorphic. Like RSA, some are multiplicatively homomorphic such as the scheme proposed by ElGamal [6]. Others will be additively homomorphic, such as Benaloh's [4] and Paillier's [5] cryptosystems.

It wouldn't be until 2009 that the first fully homomorphic encryption scheme would be proposed. For his dissertation [7] Craig Gentry used ideal lattices to implement a fully homomorphic scheme. With this discovery it is no longer necessary to choose addition or multiplication when modifying ciphertexts, algorithms that utilize both can now be applied effectively.

2.2 From Theory to Application

In 1978 Rivest *et al.* suggest that the "privacy homomorphism" may someday find use in banking transactions. [3] In 1987 Goldreich *et al.* proposed a method using oblivious transfer [8] generalized to play any of a certain class of game. [9] Turing-machine games with incomplete information played over a distance rely on a trusted third party to record either the moves made by players or game state. By using oblivious transfer players are able to pass updates of the game state or move record without knowing the current state. This system works if greater than half of the players are honest-but-curious; a term used to describe players that will try to learn any information possible, but will not lie about

their move, or successfully break encryption. This cryptosystem did not gain much traction in the gaming industry as company servers made an effective trusted third party without the need for so much computational overhead.

Despite never gaining much traction in gaming, this cryptosystem did provide a framework for computation between multiple parties with preservation of privacy. A problem that remains with this system is the requirement for a majority of non-malicious participants; not something that can be guaranteed in adversarial situations such as gaming. However, this system does present a perfect utility for cooperative situations; such as when collaboration and honesty are mutual benefits but the parties do not wish to risk exposing protected data.

2.3 Collaborative Classification

One application of mutually beneficial computation is described by Lindell and Pinkas. [10] They build upon the method generalized by Goldreich *et al.* [9] but instead of using the method to convey moves in a game requiring secrecy, the method proposed allows two parties to train a decision tree model on two sets of data without the need to expose their data to each other, or relying on a trusted third party. The classifier that they use to train their decision tree model is the ID3 algorithm invented by Ross Quinlan. [11] The suggested application for this research is by financial institutions make lending more accurate decisions without exposing private customer financial information.

ID3(R,C, T)

1. If R is empty, return a leaf-node with the class value of the majority of the transactions in T.
2. If T consists of transactions with all the same value c for the class attribute, return a leaf-node with the value c (finished classification path).
3. Otherwise,
   (a) Find the attribute that *best* classifies the transactions in T, let it be A.
   (b) Let $a_1, \ldots, a_m$ be the values of attribute A and let $T(a_1), \ldots, T(a_m)$ be a partition of T s.t. every transaction in $T(a_i)$ has the attribute value a_i.
   (c) Return a tree whose root is labeled A (this is the test attribute) and has edges labeled $a_1, \ldots, a_m$ such that for every i, the edge a_i goes to the tree ID3$(R - \{A\}, C, T(ai))$.

Figure 3: The ID3 Algorithm for Decision Tree Learning. [10, p. 4]

Collaborative training works well for situations in which both parties wish to make use of the same classifier, are both in possession of data they do not wish to disclose to the other, and both have the computational resources necessary to train the model. Often these three factors are not the case and for those situations it is likely better to use a Client-Server method.

2.4 Client-Server Classification

Barni *et al.* [12] develop a client-server style system for classification. They use the Paillier cryptosystem [5]. Additionally, they use an oblivious transfer method similar to Goldreich *et al.* [9], and Yao's garbled circuits. [13] For classification their system builds linear branching programs, a non-binary version of the decision tree algorithm used by Lindell and Pinkas. [10]

$$abs(\mathbf{a}_i^\ell \circ \mathbf{x}^\ell) = abs(\sum_{j=i}^{n} a_{i,j}^\ell x_j^\ell) \leq \sum_{j=1}^{n} 2^{2(\ell-1)} = n2^{2(\ell-1)}$$
$$\Rightarrow \ell' = 1 + \lceil \log_2(n2^{2(\ell-1)}) \rceil = 2\ell + \lceil \log_2 n \rceil - 1$$

Figure 4: Using the attribute vector $\mathbf{x}^\ell$ and linear combination vector $\mathbf{a}_i^\ell$ to determine the "bit-length ℓ' of threshold values $t_i^{\ell'}$" [12, p. 6]

They propose the use of their algorithm to operate on medical data, the representative example given is one concerning classification of ECG waveforms, and is a great example of how client-server classification may be used in general;

> A patient (client $\mathcal{C}$) owns an ECG signal and asks a service provider (server $\mathcal{S}$) to determine which class the ECG signal belongs to. $\mathcal{C}$ requires $\mathcal{S}$ to gain no knowledge about the ECG signal (as this is sensitive personal data of $\mathcal{C}$), whereas $\mathcal{S}$ requires no disclosure of details of the classification algorithm to $\mathcal{C}$ (as this represents valuable intellectual property of $\mathcal{S}$). [12, p. 13]

With several proofs of concept and different approaches to secure classification Bost *et al.* [14] seek to construct a framework to provide tools helpful to future researchers and developers doing similar work. They build their framework with several options for cryptosystems. They include Paillier's cryptosystem [5], the quadratic residuosity cryptosystem developed by Goldwasser and Micali [15], and the open source fully homomorphic HELib framework [16]. The building blocks that they provide are a secure method of comparison, a secure method for determining an argument's maximum, a method for changing the encryption scheme of the data (e.g. from Paillier to HELib), a method for calculating dot-products, and a method of dealing with floating point values used in classifiers. To demonstrate the usability of these building blocks [14] build three different classifiers.

$$v_i = (p_2) \cdot 2^{\delta_i} \cdot 2^{\min_i e_i - 52}$$

Integer representation of probability v_i is determined from the double-precision floating point p_2, precision e_i, and difference between current and minimum precision δ_i .

Figure 5: Refactorization of the float method from [14, p. 8]

"Private hyperplane decision" is trained using the private argument maximum finder they name ARGMAX and the dot product method. It functions very similarly to a binary decision tree model.

$$p(C = c_i, X_1 = x_1, \cdots, X_d = x_d) = p(C = c_i) \sum_{j=1}^{d} p(X_j = x_j | C = c_i),$$

Figure 6: [14, p. 4] Naïve Bayes classifier refactorization (Assumes independent variables)

"Secure naïve-Bayes" uses ARGMAX and the floating point method. The floating point method converts the value into a large integer; this relies on Paillier's 2^{1024} bit message space. With these two functions the algorithm is able to generate probability tables usable for Bayesian classification.

"Private decision tree" is a polynomial decision tree model. This algorithm uses HELib's fully homomorphic encryption set to a fixed-depth. Fixing the depth limits the iterative multiplicative depth homomorphism of the ciphertext and decreases the compute time. This algorithm also leverages single input, multiple data (SIMD) parallelism to further reduce compute time. When compared to the algorithm presenting in [12], Bost *et al.* are able to return classification results approximately three times faster.

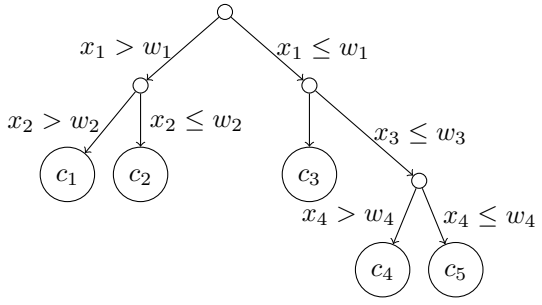

Figure 7: [14, p. 4] Binary Tree Example

Li *et al.* [17] continue work with this framework and expand it to process data from multiple sources. From the framework they use the Paillier cryptosystem [5] and the secure naïve-Bayes algorithm [14]. In order to allow the system to train over multiple data sources they provide options for both horizontal and vertical partitioning of data. Additionally, they rely on a semi trusted third party for data aggregation. Specifically, this is a third party in addition to the data providers (clients) and the party performing the analysis (server). Two public key pairs are generated, $< pk_1, sk_1 >, < pk_2, sk_2 >$. Both public keys are given to the data owners, the data collector may apply sk_1 then performs classifier training utilizing the data still protected by pk_2. The model generated may then be passed to the data receiver, and finally decrypted with sk_2. A major weakness in this scheme is shown here as it relies on the third party to provide the computational resources to actually train the model.

$$
\begin{aligned}
&\tau_{1,0} \cdot c_i^{(1,1)} \cdot \ldots \cdot c_i^{(1,s)} \cdot c_i^{(Y)} \\
=&\tau_{1,0} \cdot \prod_{k=1}^{s} \left(\tau_{1,k} \cdot \left[\left[n_i^{(k)} \cdot l \right] \right]_1 \right) \cdot (\tau_{1,s+1} \cdot [[y_i \cdot l]]_1) \\
=&\prod_{k=0}^{s+1} \tau_{1,k} \cdot \left[\left[y_i \cdot l + \sum_{k=1}^{s} (n_i^{(k)} \cdot l) \right] \right]_1 \\
=&[[y_i \cdot l + n_i \cdot l]]_1 \\
=&[[n_i' \cdot l]]_1 \mod N_1^2
\end{aligned}
$$

τ : Factor
e : Encrypted Factor
l : Integer Transforming Value
y : Noise Value
n : Number of Samples
N : Paillier Modulus

Figure 8: Horizontal aggregation method presented in [17]

Goa *et al.* [18] also extend the [14] framework but focus on increasing resistance to substitution-then comparison attacks after the model has been generated. They use the same naïve-Bayes classifier, and opt for Paillier's cryptosystem as "[i]t does not use fully homomorphic encryptions and thus it is more efficient than [other]'s proposals which rely on FHE." [18, p. 3] To achieve the increased resistance to STC attacks they present a novel technique that they label "double-blinding". Double-blinding as they describe it is essentially Rabin's oblivious transfer technique [8] applied in a manner that aptly obscures both client and server. This is accomplished by sending keys to the user. The user encrypts with provided keys and an additive homomorphism factor. The server classifies both and returns to user. The user then uses a secure comparison protocol to determine which is the correct ciphertext. The user will then decrypt the correct ciphertext for their classification result. The overhead of this implementation scales linearly and is primarily borne on the server side.

Sadegh-Riazi *et al.* [19] develop a framework similar to Bost *et al.* in [14] but remove the flexibility of the building blocks, optimizing for specifically the Naïve-Bayes classifier. By limiting the scope they are able to propose an algorithm that is extremely similar and slightly better optimized. They also compare their implementation to commercial and open source options with favorable results.

Layer Type	**Functionality**
Fully Connected (FC)	$x_i^{(L)} = \Sigma_{j=0}^{N_{L-1}-1} W_{ij}^{L-1} \times x_j^{L-1}$
Activation Layer (Act)	$x_i^{(L)} = f(x_i^{(L-1)})$
Convolution Layer (C)	$x_{ij}^{(L)} = \Sigma_{a=0}^{s_q-1} \Sigma_{b=0}^{s_q-1} W_{ab}^{L-1} \times x_{(i \cdot s_t+a)(j \cdot s_t+b)}^{L-1}$
Mean-Pooling (MeP)	$x_{ij}^{(L)}$ =Mean$(x_{(i+a)(j+b)}^{(L-1)})$, $a, b \in \{1, 2, \ldots, s_q\}$
Max-Pooling (MaP)	$x_{ij}^{(L)}$ =Max$(x_{(i+a)(j+b)}^{(L-1)})$, $a, b \in \{1, 2, \ldots, s_q\}$

Table 1: Deep and Convolutional Deep Neural Network layer types [19, p. 9]

Yasumura *et al.* [20] propose an algorithm that is nearly identical to the one proposed by Li *et al.* [17] but without "double-blinding" and replacing the hybrid cryptosystem with a single fully homomorphic implementation. They claim a higher efficiency than Li *et al.* based on "their proxy re-encryption is unnecessary as the clients can encrypt their data using the system's common public key instead of using their own set of keys. Thus, while their proxy re-encryption lasts from a few seconds to a few minutes in addition to the actual classification of data, our classification protocol completes in approximately 3.3 s for a four-class classifier." [20, p. 8] Though it should be noted that they did not actually test an implementation of the Li *et al.* algorithm.

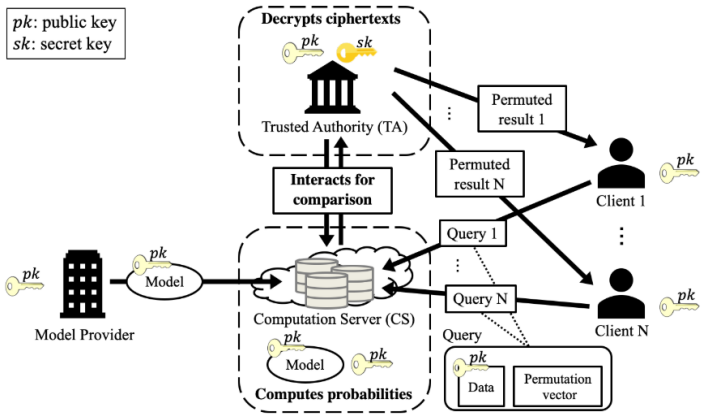

Figure 9: Visual Overview of system by Yasumura *et al.* [20, p. 5]

3 Conclusion

With the surge of research into Machine Learning after 2015 [21] the desire for security is well met by homomorphic cryptosystems. Earlier major advances in cryptography provided researchers with a strong body of literature and many tools to work with. Improvements in encryption will likely improve the efficiency or security of future algorithms; but are not likely to fundamentally change the interaction between cryptography and machine learning. Any major change in the interaction will likely be the result of novel changes in machine learning. Advances in hardware optimized for machine learning may not fare well with current cryptosystems. At the time of this writing we were unable to find any research addressing encryption of operations performed on tensor-processing units (TPUs).

The increasing sophistication of the schemes proposed by the researchers above have made introducing machine learning into secure contexts a more viable option. Of the literature reviewed here the use-cases proposed have been to the finance and medical industries, in both cases the value is increased through collaboration from different sources and security is required. Here we suggest that there are many more use cases available.

It may be beneficial to smaller entities to rent computation from cloud service providers to develop their own models without the necessity of exposing proprietary information. Likewise, it may be beneficial for an organization to utilize secure methods in classification to improve their own cyber-defense posture. Logically separating the data store and the processing resources gives the organization an opportunity for an additional layer to their defense in depth. In this manner an organization may also separate data into multiple classification levels and leverage still the same compute resources. This ability to compartmentalize data processing will allow an organization more flexibility in scaling their resources to meet their needs, and to increase the efficiency of how they use the resources already in place.

References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, p. 120–126, Feb. 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[3] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," 1978.

[4] J. Benaloh, "Dense probabilistic encryption," in *Proceedings of the workshop on selected areas of cryptography*, 1994, pp. 120–128.

[5] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[6] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[7] C. Gentry, "A Fully Homomorphic Encryption Scheme," *Dissertation*, no. September, p. 169, 2009. [Online]. Available: http://cs.au.dk/{~}stm/local-cache/gentry-thesis.pdf

[8] M. O. Rabin, "How to exchange secrets with oblivious transfer." *IACR Cryptol. ePrint Arch.*, vol. 2005, no. 187, 1981.

[9] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computingS*, no. January, pp. 218–229, 1987.

[10] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Annual International Cryptology Conference*. Springer, 2000, pp. 36–54.

[11] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[12] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A. R. Sadeghi, and T. Schneider, "Secure evaluation of private linear branching programs with medical applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5789 LNCS, no. July 2015, pp. 424–439, 2009.

[13] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[14] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine Learning Classification over Encrypted Data," in *NDSS*, vol. 4324, 2015, p. 14.

[15] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '82. New York, NY, USA: Association for Computing Machinery, 1982, p. 365–377. [Online]. Available: https://doi-org.ezproxy.uis.edu/10.1145/800070.802212

[16] S. Halevi. (2013) Helib - an implementation of homomorphic encryption. Accessed: 2020-07-06. [Online]. Available: https://github.com/shaih/HElib

[17] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018. [Online]. Available: https://doi.org/10.1016/j.ins.2018.02.056

[18] C. z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Information Sciences*, vol. 444, pp. 72–88, 2018. [Online]. Available: https://doi.org/10.1016/j.ins.2018.02.058

[19] M. Sadegh Riazi, E. M. Songhori, C. Weinert, T. Schneider, O. Tkachenko, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," *ASIACCS 2018 - Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*, pp. 707–721, 2018.

[20] Y. Yasumura, Y. Ishimaki, and H. Yamana, "Secure naïve bayes classification protocol over encrypted data using fully homomorphic encryption," *ACM International Conference Proceeding Series*, 2019.

[21] R. Perrault, Y. Shoham, E. Brynjolfsson, J. Clark, J. Etchemendy, B. Grosz, T. Lyons, J. Manyika, S. Mishra, and J. C. Niebles, "The ai index 2019 annual report," *AI Index Steering Committee, Human-Centered AI Institute, Stanford University, Stanford, CA*, 2019.