Homomorphic Encryption Application on FinancialCloud Framework

Hsin-Tsung Peng*, William W.Y. Hsu†‡§, Jan-Ming Ho§, and Min-Ruey Yu†
*Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan 106
†Department of Computer Science and Engineering
National Taiwan Ocean University, Keelung, Taiwan 202
Email: wwyhsu@ntou.edu.tw
‡Correspondence Author
§Institute of Information Science
Academia Sinica, Nankang, Taiwan 116

***Abstract*—Data security and privacy is a major concern for the users while using software services on the cloud. When users want to compute on a cloud service, traditional encryption schemes can be applied to encrypt and transfer the data to the cloud service. However, the service provider must decrypt the data for input into their computational model and thus the data content is exposed. If users do not want service providers to know what they are computing, then computing on encrypted data preserving privacy is an important issue. Homomorphic encryption is an encryption method where computations can be performed on the ciphertext, and the decrypted result of these computations is the same as if the computations were performed on the plaintext. However, the performance of this approach is currently inefficient. This paper presents an application of homomorphic encryption method on an open financial cloud framework (FinancialCloud) to perform calculations on encrypted data, therefore securing the data throughout the whole process. We demonstrate by example, showing that by applying improved algorithms can lessen the deficiencies induced by homomorphic encryptions.**

***Index Terms*—Homomorphic encryption, FinancialCloud, Cloud computing, Options pricing**

I. Introduction

In financial industry, cloud computing has become a potential solution since it provides the cost-saving, the mobility, the time-to-market and the scalability characteristics. Recently, there are cloud-based commercial software products and researches which involving financial services from a desktop model to cloud infrastructure. Chang et al. [1] presented cloud platforms that integrate Financial Software as a Service (FSaaS) and the IBM Fine Grained Security Framework. They demonstrate how portability, speed, accuracy, reliability, and security can be achieved when hosting financial services on clouds. Peng et al. also presented that derivative pricing process can be modularized and standardized [2] and cloud computing could deal with complex derivative pricing and risk management for real-time reporting limitation [3]. Misys [4], a global company delivering application software and services for banking, treasury, and capital markets, collaborates with Microsoft. They target banks with an integrated platform for portfolio management, risk management, and financial derivative valuation using the Windows Azure cloud platform. FINCAD [5], one of the largest financial service companies in the world, uses Software as a Service (SaaS) model to accurate financial reporting and necessary accounting disclosures and lowers the total cost of customers. Pricing Partners [6], a part of Thomas Reuters, is a service provider of derivative pricing and offers SaaS platform a simple procedure to achieve derivative portfolios valuation.

The SaaS is a new-style software delivery model in the cloud environment and has become a common model for many business applications. In the SaaS model, the users personal data are deployed by the service providers who maintain or use it to evaluate both the value and risk of derivatives. When the data are transferred from one service to another, we use data encryption mechanisms to enforce the privacy of data. However, it is necessary to decrypt the encrypted data because the service can only do computations on the plain data, which affect the confidentiality of data hosted in the cloud. Service providers can be non-trustworthy, users personal data will face leakage risks. For example, they may disseminate or sell the data to the competitors. This has led to increasing concerns about the privacy of the data. To prevent malicious service providers from disseminating users personal data, designing methods which continue to be effective without compromising privacy is necessary.

Homomorphic encryption mechanism allows specific types of computations to be carried out on ciphertext and obtains an encrypted result which, when decrypted, matches the result of operations performed on the plaintext [7], [8]. In other words, it allows cloud services to operate on encrypted data without knowing the original plaintext. It has been used for supporting simple aggregations, numeric computations on encrypted data as well as for private information retrieval. Homomorphic encryption includes two different types of homomorphism: the multiplicative encryption scheme and the additive encryption scheme. For example, the encryption schemes used by RSA is multiplicative and Paillier is additive. An encryption scheme is additive if:

$$\epsilon(x+y) = \varepsilon(x) \Theta \varepsilon(y), \tag{1}$$

Alternatively, an encryption scheme is multiplicative if:

$$\epsilon(x \cdot y) = \varepsilon(x)\Theta\varepsilon(y), \tag{2}$$

where ε denotes an encryption function, and Θ denotes an operation depending on the two plaintexts x and y. There are two different homomorphic cryptosystems, i.e., fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE). PHE supports only single operation on the ciphertext. This operation can be additive or multiplicative. Alternatively, FHE supports both additive and multiplicative operations. There are several existing applications of homomorphic encryption in cloud-based applications [9]. Ahmad et al. proposed a method to perform the operations on encrypted data by combing proxy re-encryption technique to prevent ciphertext from chosen ciphertext attack (CCA) [10]. In the health monitoring industry, Kocabas et al. provided a cloud-based application to a long-term patient ECG-data monitoring system [11]. To keep the user data secure and confidential in a cloud environment, Brenner et al. presented a method to compute a secret program on an untrusted server by using fully homomorphic encryption [12]. López-Alt et al. constructed a multi-key FHE scheme that can operate on encryptions under different and unrelated public keys to enhance the level of information security since the traditional FHE schemes operate on ciphertext under the same key [13]. Abe and Suzuki [14], Yokoo and Suzuki [15], and Suzuki and Yokoo [16] proposed price auction schemes to offer secrecy of bidding price in combinatorial auctions by using homomorphic encryption.

In this paper, we combine the homomorphic encryption scheme with the FinancialCloud [2], an open cloud framework for derivative pricing that can bring service providers together. FinancialCloud performs a collaborative derivative pricing by combining various cloud services since the service providers can deliver their services into this cloud framework. The approach in this research is to encrypt the data before sending to the pricing services, and pricing services perform computation on encrypted data without decrypting them. It enforces better data security which information is relayed between services. Using the binomial option pricing model (BOPM) as an example, we demonstrate the availability of homomorphic encryption. Since the efficiency of homomorphic encryption is slow, we proposed an improved algorithm of Cox-Ross-Rubinstein binomial tree [17] to improve its efficiency.

This paper is organized as follows. In Section II, we describe the proposed methodology of European option pricing by using the CRR binomial tree developed by Cox, Ross, and Rubinstein. In Section III, we demonstrate the experiment results and the usefulness of homomorphic encryption. Finally, we summarize our conclusions and indicate the direction of our future work in Section IV.

II. Methodology

This research combines the homomorphic encryption scheme with the FinancialCloud, and the goal is to force pricing services to compute on an encrypted domain. Pricing services must perform computation on encrypted data without decrypting them. It ensures improved data security in the communication between services.

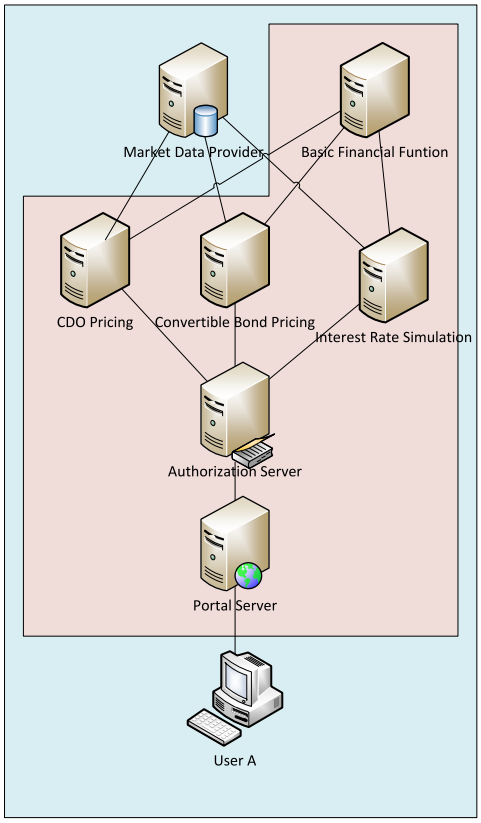

Fig. 1. System architecture of FinancialCloud.

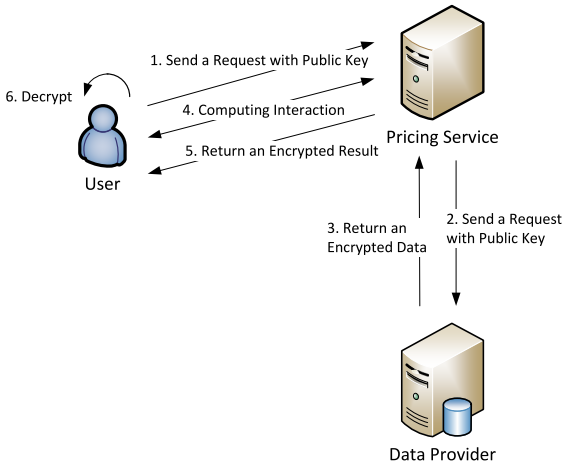

Fig. 2. Flowchart of the homomorphic encryption mechanism.

FinancialCloud is an open cloud framework for derivative pricing that brings service providers, including pricing models, algorithms, and market data together. Figure 1 depicts the system architecture of FinancialCloud. The service providers can deliver their services into the cloud framework by registering at the authorization server of the cloud. The users can easily

integrate and host modularized financial services on demand to meet users needs. The details of this cloud framework can be found in [2].

Figure 2 illustrates the homomorphic encryption mechanism. The user generates both private key and public key from a specific encryption scheme. Following, the user sends a request with a public key to pricing service, and pricing service sends another request with a public key to the data provider for gathering the desired data. Data provider encrypts the raw data by using the public key then returns to pricing service. Finally, pricing service performs the calculation based on encrypted data. Although fully homomorphic encryption (FHE) supports arbitrary computations on encrypted data, some kinds of numeric computation were not discussed and implemented in recent studies, e.g., inequality and probability-based computations. In this paper, we apply the data interaction between user and pricing service to perform these kinds of computation. After receiving the encrypted numbers, the user uses a private key to decrypt them, performs computation, and then returns the encrypted result to pricing service. Finally, the user gets the encrypted result from pricing service and decrypts it.

A. European Option Pricing using Binomial Tree

In this paper, we use the binomial tree developed by Cox, Ross, and Rubinstein to price European options [17]. The binomial tree assumes that at each time interval, the underlying stock price can only go up or down. Let u be the ratio of up movement for a stock after each interval and let d be the ratio of down movement after each interval. Then the values of u and d are:

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta t}} \\ d &= e^{-\sigma\sqrt{\Delta t}}, \end{aligned}$$

where σ is the volatility of the underlying, and Δt is the ratio of the time period of one interval to the time period for which is calculated. The probability of the stock price moving higher is:

$$p = \frac{e^{r\Delta t} - d}{u - d}, \tag{3}$$

where r is the risk-free interest rate. Then, we can obtain the value of each possible stock price at all of the time points in the binomial tree as shown in Figure 3. Note that S is the current price (spot price) of underlying. The price of a European call and put option at each final node can be computed by the equation below:

$$C_{(n,i)} = \text{MAX}(S_{(n,i)} - K, 0), \tag{4}$$

for a call option and

$$C_{(n,i)} = \text{MAX}(K - S_{(n,i)}, 0), \tag{5}$$

for a put option, where K is the strike price, $C_{(n,i)}$ and $S_{(n,i)}$ is the option value and the spot price of underlying for the i-th node at time n, respectively. Once the terminal conditions are established, the expectation value of option is found for each final node, starting at the penultimate time step, and working back to the first node of the tree by the following equation:

$$C_{(t-\Delta t,i)} = e^{-r\Delta t} \cdot (p \cdot C_{(t,i+1)} + (1-p) \cdot C_{(t,i-1)}), \tag{6}$$

where $C_{(t-t,i)}$ is the expectation value of option for the i-th node at time t.

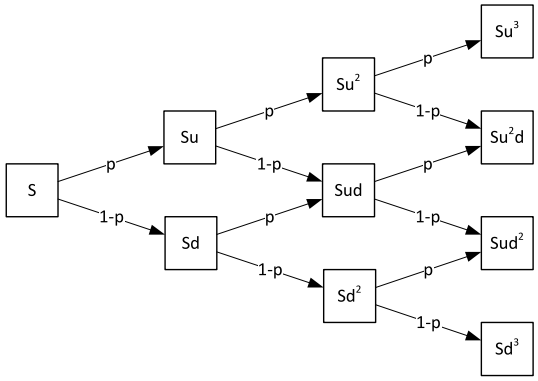

Fig. 3. Binomial tree with height equal to 3.

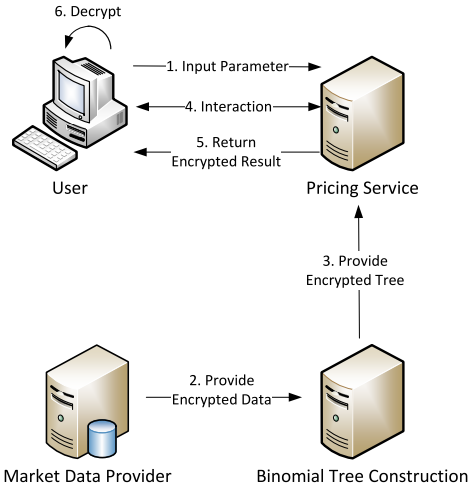

Fig. 4. Flowchart of European options pricing.

B. Binomial Option Pricing with Homomorphic Encryption

Figure 4 shows the flowchart of pricing European option using homomorphic encryption. In the FinancialCloud, the collaborative procedure is composed of three services, the "Market Data Provider", the "Binomial Tree Construction" and the "Pricing Service". We apply the data interaction between user and pricing service to perform the inequality, the multiply and the division computations due to the limitation of recent studies as mentioned before. The "Pricing Service" sends two encrypted numbers (one is the exercise value, the other one is the expectation value of option) to the user for performing inequality computation when it needs to calculate the value of an option at the expiration date. After receiving the encrypted numbers, the user uses the private key to decrypt them, performs computation, and then returns the encrypted

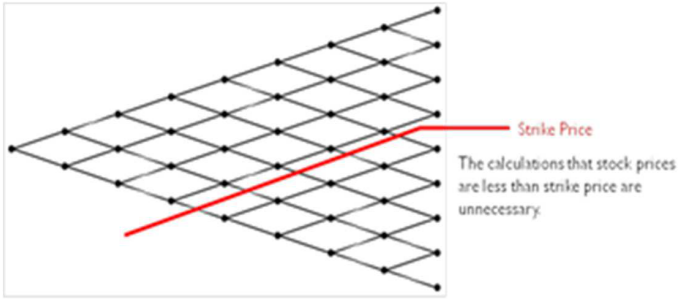

Fig. 5. Improved binomial options pricing algorithm.

result to pricing service. Finally, the user gets the encrypted result from pricing service and decrypts it.

C. Improved Binomial Option Pricing with Homomorphic Encryption

Since the efficiency of homomorphic encryption is slow, we proposed an improved binomial algorithm for speeding up the option pricing procedure. Figure 5 shows the illustration of improved algorithm. Using the European call option as an example, the calculations, of which the spot prices of nodes are lower than the given strike price, are unnecessary, and thus the efficiency can be improved by pruning these nodes.

III. Experiments

We demonstrate the usefulness of homomorphic encryption mechanism by using the European option. The binomial tree algorithm is used to price the option, and the parameter settings are listed in Table I.

We used a PHE open source library, which supports floating-point number computation and use Python to implement the whole services. The computing environment is dual 2.93GHz cores with a 2.00 GB RAM, and the operating system is Window Server 2008 R2. We set the heights of the binomial tree from 10 to 180, i.e., the numbers of node range from 55 to 16,290, and each set runs 100 iterations to obtain average computational cost. Moreover, the elapsed computing time excluded the communication between the user and the services.

Figure 6 and 7 shows the comparison of computational cost over the heights of the binomial tree, and Table II shows the experiment results. We can see that homomorphic encryption mechanism increases computational time drastically, e.g., as shown in Figure 6 and Figure 7, the option pricing with homomorphic encryption is slower than the one without homomorphic encryption (0.014 versus 1835.17) when we fixed the height of the binomial tree to 180. The improved algorithm, shown in Figure 7, gets better computation performance than homomorphic encryption (1375.59 versus 1835.17).

IV. Conclusion

Cryptography lacks practical works especially in the fields of financial computing, and this paper is one of the few research that addresses encryption in derivative pricing process

TABLE I
Parameters of the option for the experiment.

Parameter	Value
Spot Price	100
Strike Price	100
Risk-free Interest Rate (%)	1
Underlying Volatility (%)	12.5
Expiration Time (Annualized)	0.5
Heights of the Binomial Tree	From 10 to 180, step size 10

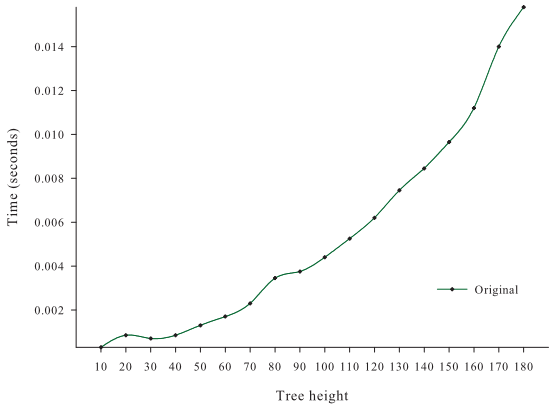

Fig. 6. Normal options pricing without homomorphic encryption.

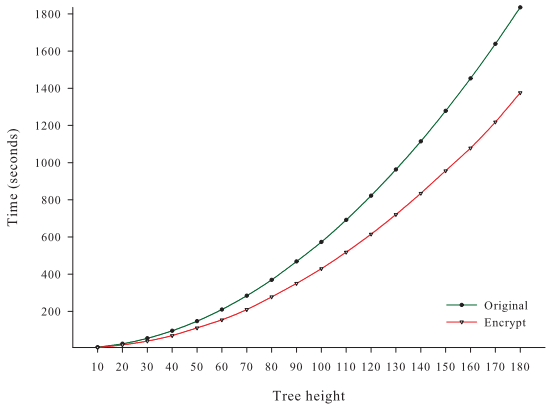

Fig. 7. Options pricing with homomorphic encryption.

on the cloud. This paper combines the homomorphic encryption with FinancialCloud, encrypting the data before sending to the pricing services. The pricing services perform computation on an encrypted domain, ensuring better data security in the communication between services. By demonstrating the availability of homomorphic encryption using the binomial tree to price European options, we show that our approach is doable. Since the efficiency of homomorphic encryption is slow, we proposed an improved algorithm of the binomial tree for speeding up the calculation. Privacy preservation is one of the many important issues in cloud computing, and many related research to preserve privacy have been

TABLE II
EXPERIMENT RESULT OF THE HOMOMORPHIC ENCRYPTION MECHANISM. THE EUROPEAN CALL OPTION PRICING WITHOUT HOMOMORPHIC ENCRYPTION MECHANISM IS NOTED AS "ORIGINAL", THE ONE WITH HOMOMORPHIC ENCRYPTION MECHANISM IS NOTED AS "ENCRYPT", AND THE OTHER ONE WITH IMPROVED ALGORITHM IS NOTED AS "IMPROVED".

Height of Binomial Tree	Numbers of Nodes	Original (s)	Encrypt (s)	Improved (s)
10	55	0.00030	7.08	5.54
20	210	0.00085	25.25	19.35
30	465	0.00070	54.62	39.68
40	820	0.00085	95.09	69.54
50	1,275	0.00130	146.78	110.74
60	1,830	0.00170	209.42	154.26
70	2,485	0.00230	283.60	209.17
80	3,240	0.00345	368.86	277.18
90	4,095	0.00375	467.74	350.06
100	5,050	0.00440	572.62	429.31
110	6,105	0.00525	691.49	518.11
120	7,260	0.00620	821.81	614.92
130	8,515	0.00745	962.61	720.56
140	9,870	0.00845	1113.85	834.49
150	11,325	0.00965	1278.19	956.19
160	12,880	0.01120	1452.55	1078.22
170	14,535	0.01400	1638.49	1218.24
180	16,290	0.01580	1835.17	1375.59

proposed to prevent the disclosure of data. In this paper, we apply a privacy preserving mechanisms using homomorphic encryption mechanism. We have taken a starting step, and the privacy preserving mechanism needs to be further improved. In future research, the following directions will be considered:

1) RSA-based and Paillier-based Homomorphic encryption mechanisms are vulnerable to chosen ciphertext attack (CCA). To prevent encrypted data from CCA, the cloud framework should provide a kind of proxy re-encryption algorithms to achieve good privacy preservation.
2) It is necessary to speed up the computation of homomorphic encryption since the exotic derivatives are more complex than the vanilla options as shown in this paper.
3) Financial applications will require the implementation of several functions, such as the standard deviation, inequality, logistic regression and probability-based calculations, as discussed by [9], [18].

ACKNOWLEDGMENT

The authors are supported by the Ministry of Science and Technology of Taiwan, under grants MOST-105-2221-E-019-062, MOST-104-2221-E-019-050, MOST-102-2221-E-001-015-MY3, and MOST-105-2221-E-019-062. This research has no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

REFERENCES

[1] V. Chang, C.-S. Li, D. De Roure, G. Wills, R. J. Walters, and C. Chee, "The financial clouds review," *Cloud Computing Advancements in Design, Implementation, and Technologies*, vol. 125, 2012.

[2] H.-T. Peng, W. W. Hsu, C.-H. Chen, F. Lai, and J.-M. Ho, "FinancialCloud: Open cloud framework of derivative pricing," in *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 2013, pp. 782–789.

[3] H.-T. Peng, C.-F. Chang, S.-L. Liao, M.-Y. Kao, F. Lai, and J.-M. Ho, "The development of a real-time valuation service of financial derivatives," in *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 2012, pp. 1–8.

[4] (2016) Misys. [Online]. Available: http://www.misys.com

[5] (2016) FiNCAD. [Online]. Available: http://www.fincad.com

[6] (2016) Pricing partners. [Online]. Available: http://pricingpartners.com

[7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, ser. STOC '09. New York, NY, USA: ACM, 2009, pp. 169–178. [Online]. Available: http://doi.acm.org/10.1145/1536414.1536440

[8] C. Gentry and S. Halevi, *Implementing Gentry's Fully-Homomorphic Encryption Scheme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 129–148. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20465-4_9

[9] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, ser. CCSW '11. New York, NY, USA: ACM, 2011, pp. 113–124. [Online]. Available: http://doi.acm.org/10.1145/2046660.2046682

[10] I. Ahmad and A. Khandekar, "Homomorphic encryption method applied to cloud computing," *International Journal of Information & Computer Technology*, vol. 15, pp. 1519–1530, 2014.

[11] O. Kocabas, T. Soyata, J.-P. Couderc, M. Aktas, J. Xia, and M. Huang, "Assessment of cloud-based health monitoring using homomorphic encryption," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*. IEEE, 2013, pp. 443–446.

[12] M. Brenner, J. Wiebelitz, G. Von Voigt, and M. Smith, "Secret program execution in the cloud applying homomorphic encryption," in *5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011)*. IEEE, 2011, pp. 114–119.

[13] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 1219–1234.

[14] M. Abe and K. Suzuki, "M+ 1-st price auction using homomorphic encryption," in *International Workshop on Public Key Cryptography*. Springer, 2002, pp. 115–124.

[15] M. Yokoo and K. Suzuki, "Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM, 2002, pp. 112–119.

[16] K. Suzuki and M. Yokoo, "Secure generalized Vickrey auction using homomorphic encryption," in *International Conference on Financial Cryptography*. Springer, 2003, pp. 239–249.

[17] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option pricing: A simplified approach," *Journal of financial Economics*, vol. 7, no. 3, pp. 229–263, 1979.

[18] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Annual Cryptology Conference*. Springer, 2011, pp. 505–524.