ON PRIVACY PREFERENCE IN COLLUSION-DETERRENCE GAMES FOR SECURE MULTI-PARTY COMPUTATION

Zhaohong Wang and Sen-ching S. Cheung

Department of Electrical and Computer Engineering, University of Kentucky, USA

ABSTRACT

Secure multi-party computation (MPC) has been established as the de facto paradigm for protecting privacy in distributed computation. Information-theoretic secure MPC protocols, though more efficient than their computationally secure counterparts, require at least three computational parties and are prone to collusion attacks. Previous work has used mechanism designs to deter collusion. An important element missing is the consideration of how different players value privacy. In this paper, we provide a detailed analysis of possible outcomes under different privacy preferences based on the relative cost of collusion attacks over loss of privacy. We explicitly calculate the conditions under which honesty is the solution. Simulation results provide further evidence to demonstrate the validity of our mechanism design.

***Index Terms*—** signal processing, encrypted domain processing, secure mpc, game theory

1. INTRODUCTION

Privacy protection in distributed computing enables distrusting parties to participate in a joint computation without revealing their secret data. As more personal information is being processed under various cloud computing platforms, there is an increasing demand for privacy protection solutions that enable collaboration without giving up valuable and sensitive data. The standard approach to protect privacy in distributed computation is to use secure multiparty computation or Secure MPC protocols. However, despite being actively researched for more than 30 years, Secure MPC protocols are still rarely used in practical systems [1].

One reason for the poor deployment of Secure MPC is its significant communication and computational costs [2]. Commonly used Secure MPC protocols, such as those based on homomorphic encryption and garbled circuits, operate on encrypted data and are secure against computationally-bound adversaries [3]. An alternative is to use the information-theoretic Secure MPC (ITS-MPC) protocols based on results in [4], otherwise known as the Ben-Or, Goldwasser and Wigderson (BGW) protocol. In these protocols, information exchanged between different parties is statistically independent of the secret data. As ITS-MPC protocols do not depend on the difficulty of specific computational problems, they often use a smaller finite field for data representations and produce more efficient application protocols [5, 6, 7].

A major disadvantage of the BGW protocol is the need to maintain a majority of non-colluding computational parties [8]. Researchers have long pointed out the danger of collusion attacks in outsourced computations [9]. In fact, collusion attacks are significant real-life problems and occur in many networked applications [10, 11]. In our earlier work, we proposed a framework to deter collusion in a BGW-based distributed computing platform by retaliation mechanisms, fake collusion attacks, and a censorship scheme [5]. The advantage of the approach proposed in [5] over other techniques is that no centralized server or computationally-intensive protocols are needed, making it ideal for high-throughput signal processing applications.

A key aspect to deter collusion is to understand the incentive behind such an attack, which is often based on the relative value of the secret over the risk of the collusion, i.e. getting caught and suffering from its consequence. Due to the heterogeneity of participants, a real world market can have very different evaluations of secrets which may lead to different outcomes. However, the analysis in [5] tackles only one specific case. In this paper, we further extend the scheme in [5] by considering the heterogeneity of the participants. We carefully analyze all possible privacy preferences, show for each case the condition under which honesty is the solution, and discuss their implications on the price for the distributed service and the strategies in growing such a market.

The rest of the paper is organized as follows: Section 2 describes the collusion attack models. Related work is discussed in Section 3. Section 4 introduces a series of the *User-Vendor* games to deter collusion attempts from the secret-data owners. Experimental results are presented in Section 5. We conclude the paper with future work in Section 6.

2. PROBLEM STATEMENT AND ATTACK MODELS

Our outsourcing computing framework comprises two types of participants: the computing platform *agents*, denoted as A_i where $i \in \{1, 2, \ldots\}$, and the platform *customers*, i.e. the secret-data owners. Denote any pair of platform customers as U (*User*) and V (*Vendor*) who want to cooperate in a joint computation. We assume that there is a coordinator C who is responsible for keeping records of the IDs of participants but does not handle any actual secret data. We focus our discussions on two parties but the scheme is general enough for an arbitrary number of parties.

U and V do not trust each other with their secret data and they do not possess the necessary CPU power for the computation. They outsource their computation to the computing platform by means of Secure MPC protocols. Despite its simplicity, this model is an abstraction of many practical scenarios. For example, in cloud computing, A_i provides platform as a service (PaaS) while U provides sensitive data and V provides proprietary software [12]. Note that our emphasis is on protecting the privacy of data, rather than the programming instructions.

This work was supported in part by the National Science Foundation under Grants 1018241, 1444022, and 1237134.

Imagine an outsourced computation is built from the BGW protocol. We need at least $n = 3$ agents to carry out any arithmetic operations [13]. As such, *no collusion* can be tolerated since any two agents can reconstruct the secret.

Collusion, modeled as a covert adversarial behavior, can occur between one of the data owners and a portion of computing agents, or among the computing agents themselves. We can identify three types of collusion attacks. **A1: side-channel attacks**: If an adversary controls a majority of computing agents, they can exchange their secret shares freely through their pre-established side channels outside of the protocol to steal secrets. System-based approaches such as anonymous communication [14] and program obfuscation [15] can make the coordination harder but cannot provide any form of guarantee on preventing the formation of side channels. We assume that no such side channels exist among agents and model the adversaries as localized [16]. **A2: Collusion between agents and U or V**: We will focus on the agents' collusion with U, as the case for V is identical. As each agent possesses secret shares from both U and V, it is possible for U to collude with enough agents to reconstruct the secret from V. No changes in infrastructure can block such an attack as it is necessary for U to communicate with the agents. Neither can the collusion be detected from the communication between the agents and V. In Section 4, we study our mechanism between U and V to deter them from cheating. **A3: Collusion among agents**: The direct communication among agents is essential as it is needed in some procedures [17]. At the same time, direct communication opens doors for collusion. Using a star-shaped network topology, our previous work proposed a censorship scheme for U and V to destroy subliminal communication among agents [5]. This attack will not be addressed in this paper.

3. RELATED WORK

Colluding communication usually exists in two different forms: through side-channels external to the protocol, or through subliminal communication within the protocol as hidden data [18]. Algorithmically, it is impossible to design protocols to curtail communications over side-channels. Existing anti-collusion techniques focus on eliminating subliminal communication by relying on either a semi-honest/trusted centralized server or a specially-designed ballot box [19, 18, 20]. A special computer called a verifiably secure device or VSD was devised to which all players physically submit envelopes and ballot boxes [19]. In [18, 20], the mediated multiparty computation (MMPC) achieves the collusion-deterrence by means of an honest mediator who carries out a two-party secure function evaluation (SFE) with each party. All of the above approaches require heavy computation at a fortified centralized server, which defeats the efficiency goal of using BGW based ITS-MPC techniques.

General consideration of attacks on Secure MPC protocols include two different aspects: 1) how parties are identified and corrupted by an adversary, and 2) how corrupt parties behave under the control of an adversary. For the first question, existing classifications typically focus on whether an honest party becomes corrupted during the course of the computation [21]. For collusion attacks, such a question is far less important than *how* an honest party is corrupted by an adversary to collude in stealing a secret – is it due to information received in-band as part of the defined protocol or out-of-band through side channels? The first kind is termed a *local adversary* attack as it is based on local information anticipated by the protocol. The local adversarial model has been used to model collusion attacks in [16] and will be assumed in our mechanism design.

As for the second question, adversarial behaviors are typically classified into semi-honest and malicious. Assuming that there is a rational being behind each participant, there must be an external reason behind such a decision such as a higher reward or a non-negligible probability to get caught. Such behaviors are best modeled via a covert adversarial model [22]. Covert security tries to model the situations in daily life: adversaries are willing to cheat if the cheating will not be detected. There are prior works in computationally Secure MPC under the covert model [23, 24] and our proposed mechanism is an effort to extend it to ITS-MPC. The notion of covertness suggests the rational decision process of agents. There are prior studies on the use of game-theoretic construction for rational Secure MPC and rational secret sharing [25, 26]. In rational Secure MPC, each party has his/her own secret held in "hostage" by other peer parties and the goal is to encourage players to exchange shares so that all parties benefit from knowing the final answers. In contrast, our computing agents are forbidden to exchange secret shares to learn anything about the secret.

4. COLLUSION DETERRENCE GAMES

In this section, we describe the mechanisms that thwart type **A2** collusion attacks described in Section 2 formed between the computing *agents* and one of the *customers* (either U or V) to steal the other's secret.

Before starting the joint computation, there must be a legally-binding contract in place so that U and V both understand that they should not collude with agents in stealing each other's secret. Evidence collecting measures, such as a traitor tracing code, can be applied to the secret inputs to track the source of secret leakage. If one party, say V, finds out that U tried to steal V's secret, U would be liable to pay for the damages based on charges brought on by V. In retaliation, U could countercharge V with similar accusations and both parties would provide evidence to an authority for a judgment. We call this strategy undertaken by U and V *retaliation*.

With the initial strategy of staying honest or cheating and the follow-up strategy of possible retaliation, there are four possible combinations for each player or a total of 16 different interaction outcomes between them. All possible cases are listed in Table 1 with $C_U, C_V = 1$ representing cheating and $R_U, R_V = 1$ representing retaliation.

Table 1: Different Outcomes in User-Vendor Games

Case	C_U	C_V	R_U	R_V	Outcome
1	0	0	0	0	D
2	0	0	0	1	X
3	0	0	1	0	X
4	0	0	1	1	A
5	0	1	0	0	E
6	0	1	0	1	X
7	0	1	1	0	X
8	0	1	1	1	A
9	1	0	0	0	C
10	1	0	0	1	X
11	1	0	1	0	X
12	1	0	1	1	A
13	1	1	0	0	B
14	1	1	0	1	X
15	1	1	1	0	X
16	1	1	1	1	A

We mark unlikely outcomes with X based on the *an-eye-for-an-eye* assumption: if one party retaliates by filing charges for a sus-

pected cheating offense, the other party will retaliate with a counter-lawsuit, an assumption supported in real life by the large number of litigations, especially in the United States [27]. Hence, cases 2, 3, 6, 7, 10, 11, 14 and 15 are excluded from further considerations.

To study the preference ranking of these outcomes, we assume the perspective of U because the case for V is almost identical. We use a cost and benefit analysis to eliminate unlikely preference orders and model the remaining ones using different games.

The mutual retaliation always results in the least desirable outcome A. Outcome C is clearly the most desirable because U successfully steals V's secret and suffers no consequence. As V is honest, V's secret should be of high enough value to cover U's cost in collusion if this collusion attack is a rational act. All of the other outcomes are not as good. In summary, there are three preference orders we need to consider:

$$\text{C} \succ \text{D} \succ \text{E} \succ \text{B} \succ \text{A} \quad (1)$$

$$\text{C} \succ \text{D} \succ \text{B} \succ \text{E} \succ \text{A} \quad (2)$$

$$\text{C} \succ \text{B} \succ \text{D} \succ \text{E} \succ \text{A} \quad (3)$$

where the symbol $\succ$ is understood as "is preferred over". A useful way to understand these orders is based on the cost of collusion. For example, Preference (1) ranks B the lowest because the high cost of collusion exceeds even the damage of losing one's own secret in E.

As we have five outcomes to consider, we denote the normalized utility values for these outcomes as $0 = p_0 < p_1 < p_2 < p_3 < p_4 = 1$. For the three preference orders corresponding to a high collusion cost (1), medium collusion cost (2) to low collusion cost (3), the assignments of the utility values are shown in Table 2.

Table 2: Normalized Utility of Different Outcomes in User-Vendor Games

	U's Preference		
Strategies	**High cost (1)**	**Mid cost (2)**	**Low cost (3)**
Retaliate (A)	p_0	p_0	p_0
Both cheat (B)	p_1	p_2	p_3
U cheats only (C)	p_4	p_4	p_4
No one cheats (D)	p_3	p_3	p_2
V cheats only (E)	p_2	p_1	p_1

We define q as the "non-retaliating" probability for both U and V, conditioned on the other's cheating behavior. The extreme value $q = 1$ means that no one retaliates while $q = 0$ means that one always retaliates if his/her secret is stolen. We first analyze Preference (3) in the game described in the following table.

		V	
		Honest	Cheat
U	Honest	(p_2, p_2)	$(1-q)(p_0, p_0)$ $+q(p_1, p_4)$ $= (qp_1, q)$
	Cheat	$(1-q)(p_0, p_0)$ $+q(p_4, p_1)$ $= (q, qp_1)$	$(1-q^2)(p_0, p_0)$ $+q^2(p_3, p_3)$ $= (q^2p_3, q^2p_3)$

The two-tuple in each entry indicates the average payoffs of U and V when adopting the row and column strategies respectively. To make the honest strategy a solution, either (a) $p_2 > q$, or (b) $p_2 = q$ and $qp_1 > q^2p_3$, i.e. $p_1 > qp_3 = p_2p_3$, which are possible to obtain. For cheating to be a solution, either (c) $q^2p_3 > qp_1$, i.e. $qp_3 > p_1$, or (d) $q^2p_3 = qp_1$ and $q > p_2$, i.e. $p_1 = qp_3 > p_2p_3$. Note that conditions (b) and (d) are the same except that (b) requires $p_2 = q$ which is very difficult to sustain, so the situation is a mixed strategy of honesty and cheating:

$$h_u = h_v = \frac{1}{\frac{q-p_2}{q(p_1-qp_3)} + 1} \quad (4)$$

where h_u and h_v are the honest probability of U and V respectively. Similarly, for Preference (1), Honesty is achieved for both U and V if $p_3 \geq q$. When the theft is *undetectable*, i.e. $p_3 < q$, it can be shown that the following mixed strategy is the solution:

$$h_u = h_v = \frac{1}{\frac{q-p_3}{q(p_2-qp_1)} + 1} \quad (5)$$

For Preference (2), there is a general solution indicating the honest fraction as

$$h_u = h_v = \frac{1}{\frac{q-p_3}{q(p_1-qp_2)} + 1} \quad (6)$$

The above analysis serves as proof for the following summarized results.

Theorem 4.1. *For all the three preference orders outlined in (1), (2) and (3), (honest, honest) is the solution if both players being honest has strictly higher utility than the successful stealing of other's secret, i.e. $p_3 > qp_1 + (1-q)p_0 = q$ for (1) or (2), and $p_2 > q$ for (3). If the two utilities are equal, (honest, honest) is still the solution for preference order (1), but preference orders (2) and (3) require an additional condition that losing one's secret has higher utility than mutual theft, i.e. $p_1 > qp_2 + (1-q)p_0 = qp_2$ for (2) and $p_1 > qp_3 + (1-q)p_0 = qp_3$ for (3).*

Theorem 4.2. *For (2) and (3), (cheat, cheat) is the solution if mutual theft has strictly higher utility than losing one's secret, i.e. $qp_2 > p_1$ for (2) or $qp_3 > p_1$ for (3). If the two utilities are equal, (cheat, cheat) is still the solution if successful stealing has higher utility and both being honest, i.e. $q > p_3$ for (2) or $q > p_2$ for (3). (Cheat, cheat) is not a solution for (1).*

Corollary 4.3. *If none of the conditions in Theorems 4.1 and 4.2 are met, the solution is a mixed strategy.*

Theorem 4.1 is important because making p_3 high by providing and paying for high-quality services, and providing state-of-the-art theft tracking technology such as watermarking to make q small are both reasonable mechanisms in maintaining a viable market. Corollary 4.3 suggests that additional mechanisms are required to make collusion harder, such as the policing and censorship described in our earlier work [5].

5. EXPERIMENTS

In this section, we want to validate our results by simulating behaviors of a large number of customers playing the User-Vendor games in an online privacy computation market based on the GameBug simulator [30]. Similar to any large scale online markets, each customer is uncertain about others' actions and thus will attempt different strategies in order to maximize the payoff. Like the peer-to-peer streaming game in [28], we adopt the replicator dynamics (RD) to model the evolution of the group size over time.

Consider initially there are both groups of customers playing either "honesty" or "cheating". We are interested in how the population evolves under our game in choosing strategies. Denote the payoff of strategy s_i as u_{s_i}, the average payoff as $\bar{u}$, the proportion of individuals using s_i as x_i, and its rate of change as $\dot{x}_i$. The RD equation is given as $\dot{x}_i = (u_{s_i} - \bar{u})x_i$ [29]. This equation can be interpreted as follows: provided that a customer stay in the market, a larger advantage of a strategy's payoff over the average will cause

the customer to switch sooner to that strategy. As such, the growth rate of the customers using each strategy is proportional to the excess of the strategy's payoff over the average payoff.

Denote the game using the order in (1) as G1, the order in (2) as G2 and the order in (3) as G3. Each user in the system is randomly matched with a vendor from the same population for cooperations. Recall that q is the non-retaliation probability and p_i is the i-th ranked payoff. At the beginning of the simulation, 90% of the populations are honest. We first test the scenario of *G1*, with $q \leq p_3$. The initial population profile evolves very quickly toward the pure-honesty as depicted in Fig. 1. In sharp contrast, $q > p_3$ leads to a mixed profile with $h_u = h_v = 0.8$. Fig. 2 shows this evolution in the population whose profile gradually converges to the mixed profile as marked by the solid black line. Second, we test the scenario of *G2*. Under the condition $p_3 > q$, the honest strategy prevails quickly as depicted in Fig. 3. Under the alternative honest condition $p_1 > p_2 p_3$ and $p_3 = q$, however, the honest behavior evolves very slowly as depicted in Fig. 4. Both the conditions $qp_2 > p_1$ and $p_1 > p_2 p_3$ and $q > p_3$ lead the population to evolve to cheating, as depicted in Fig. 5. Third, we simulate *G3*. The honest condition $p_2 > q$ yields a relatively slow system evolution compared to the previous two games, and the condition $p_1 > p_2 p_3$ and $p_2 = q$ has a much slower evolution, as depicted in Fig. 6 and 7 respectively. The cheating conditions $qp_3 > p_1$ or $p_1 > p_2 p_3$ and $q > p_2$ have very similar effects as that of *G2*, depicted in Fig. 8.

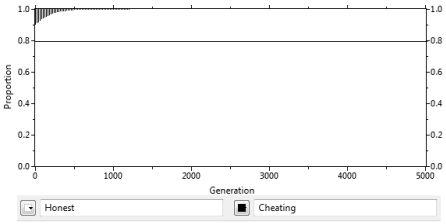

Fig. 1: Emergent Behavior in G1 when $q < p_3$

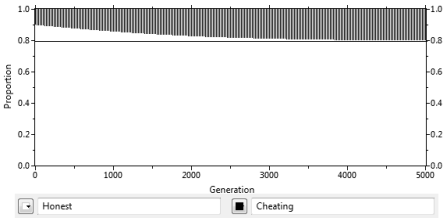

Fig. 2: Emergent Behavior in G1 when $q > p_3$

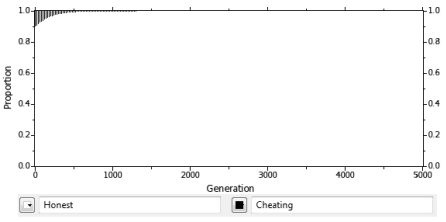

Fig. 3: Emergent Behavior in G2 when $q < p_3$

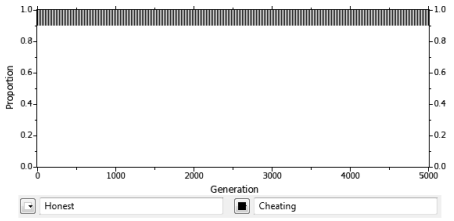

Fig. 4: Emergent Behavior in G2 when $p_1 > p_2 p_3$ and $p_3 = q$

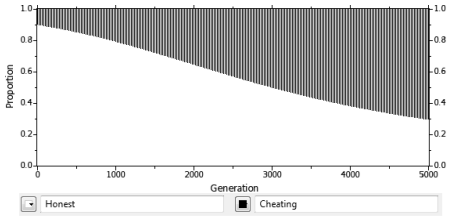

Fig. 5: Emergent Behavior in G2 when either $qp_2 > p_1$ or $p_1 > p_2 p_3$ and $q > p_3$

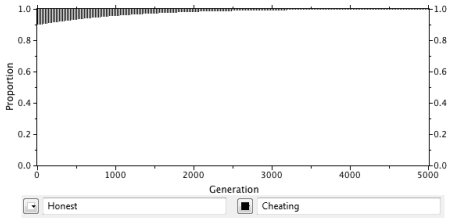

Fig. 6: Emergent Behavior in G3 when $q < p_2$

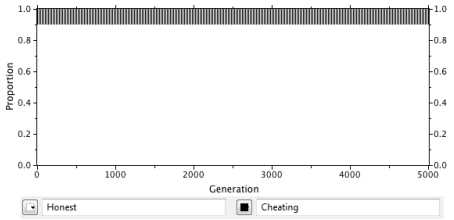

Fig. 7: Emergent Behavior in G3 when $p_1 > p_2 p_3$ and $p_2 = q$

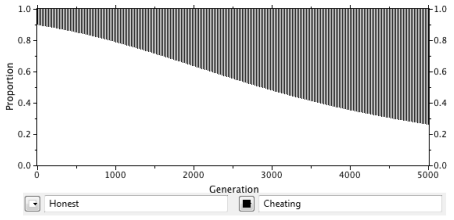

Fig. 8: Emergent Behavior in G3 when either $qp_3 > p_1$ or $p_1 > p_2 p_3$ and $q > p_2$

6. CONCLUSION

In this paper, we have demonstrated the impact of privacy preference on deterring collusion attacks in ITS-MPC frameworks via theoretical analysis and simulations. The limitation of our analysis is in the adherence to symmetric games in which both players have the same preference. We are currently extending our analysis using Bayesian Games to better model the uncertainty and asymmetry of privacy preferences in real-world scenarios.

7. REFERENCES

[1] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, et al., "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*, pp. 325–343. Springer, 2009.

[2] Ronald Cramer and Ivan Damgård, "Multiparty computation, an introduction," in *Contemporary cryptology*, pp. 41–87. Springer, 2005.

[3] S.-C. Cheung and Thinh Nguyen, "Secure multiparty computation between distrusted networks terminals," *EURASIP Journal on Information Security*, vol. 2007, 2007.

[4] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 1–10.

[5] Zhaohong Wang, Ying Luo, and Sen-Ching Cheung, "Efficient multiparty computation with collusion-deterred secret sharing," in *Acoustics, Speech, and Signal Processing, 2014 39th IEEE International Conference on (ICASSP)*. IEEE, 2014.

[6] Sayed M SaghaianNejadEsfahani, Ying Luo, and S-c S Cheung, "Privacy protected image denoising with secret shares," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 253–256.

[7] Casey Devet, Ian Goldberg, and Nadia Heninger, "Optimally robust private information retrieval," in *21st USENIX Security Symposium*, 2012.

[8] Josh Cohen Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret," in *Advances in CryptologyCRYPTO86*. Springer, 1987, pp. 251–260.

[9] Jonathan L Dautrich and Chinya V Ravishankar, "Security limitations of using secret sharing for data outsourcing," in *Data and Applications Security and Privacy XXVI*, pp. 145–160. Springer, 2012.

[10] Adrian Goldberg, "Can the world of online poker chase out the cheats?," *BBC News*, 2010, Accessed Sept. 23, 2015 at "http://www.bbc.co.uk/news/uk-11250835".

[11] Qiao Lian, Zheng Zhang, Mao Yang, Ben Y Zhao, Yafei Dai, and Xiaoming Li, "An empirical study of collusion behavior in the maze p2p file-sharing system," in *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*. IEEE, 2007, pp. 56–56.

[12] Luis M Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[13] Z. Wang and S.-C. Cheung, "Collusion deterrence in secure multiparty computation," *IEEE Transactions on Information Forensics and Security*, 2015, In preparation.

[14] Roger Dingledine, Nick Mathewson, and Paul Syverson, "Tor: The second-generation onion router," Tech. Rep., DTIC Document, 2004.

[15] Vivek Balachandran and Sabu Emmanuel, "Software code obfuscation by hiding control flow information in stack," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*. IEEE, 2011, pp. 1–6.

[16] Ran Canetti and Margarita Vald, "Universally composable security with local adversaries," in *Security and Cryptography for Networks*, pp. 281–301. Springer, 2012.

[17] Octavian Catrina and Sebastiaan De Hoogh, "Improved primitives for secure multiparty integer computation," in *Security and Cryptography for Networks*, pp. 182–199. Springer, 2010.

[18] Joël Alwen, Jonathan Katz, Yehuda Lindell, Giuseppe Persiano, Ivan Visconti, et al., "Collusion-free multiparty computation in the mediated model," in *Advances in Cryptology-CRYPTO 2009*, pp. 524–540. Springer, 2009.

[19] Sergei Izmalkov, Matt Lepinski, and Silvio Micali, "Verifiably secure devices," in *Theory of Cryptography*, pp. 273–301. Springer, 2008.

[20] Joël Alwen, Jonathan Katz, Ueli Maurer, and Vassilis Zikas, "Collusion-preserving computation," in *Advances in Cryptology–CRYPTO 2012*, pp. 124–143. Springer, 2012.

[21] Carmit Hazay and Yehuda Lindell, *Efficient secure two-party protocols: Techniques and constructions*, Springer, 2010.

[22] Yonatan Aumann and Yehuda Lindell, "Security against covert adversaries: Efficient protocols for realistic adversaries," in *Theory of Cryptography*, pp. 137–156. Springer, 2007.

[23] Isheeta Nargis, Payman Mohassel, and Wayne Eberly, "Efficient multiparty computation for arithmetic circuits against a covert majority," in *Progress in Cryptology–AFRICACRYPT 2013*, pp. 260–278. Springer, 2013.

[24] Ivan Damgård, Marcel Keller, Enrique Larraia, Christian Miles, and Nigel P Smart, "Implementing aes via an actively/covertly secure dishonest-majority mpc protocol," in *Security and Cryptography for Networks*, pp. 241–263. Springer, 2012.

[25] Joseph Halpern and Vanessa Teague, "Rational secret sharing and multiparty computation," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 623–632.

[26] John Ross Wallrabenstein and Chris Clifton, "Equilibrium concepts for rational multiparty computation," in *Decision and Game Theory for Security*, pp. 226–245. Springer, 2013.

[27] Amit Chowdhry, "Apple and samsung drop patent disputes against each other outside of the u.s.," *Forbes*, Aug. 6, 2014, accessed on Sept. 12 at "http://www.forbes.com/sites/amitchowdhry/2014/08/06/apple-and-samsung-drop-patent-disputes-against-each-other-outside-of-the-u-s/".

[28] Yan Chen, Beibei Wang, W. Sabrina Lin, Yongle Wu, and K. J. Ray Liu, "Cooperative peer-to-peer streaming: An evolutionary game-theoretic approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 10, pp. 1346–1357, 2010.

[29] Zhu Han, *Game theory in wireless and communication networks: theory, models, and applications*, Cambridge University Press, 2012.

[30] Robert Wyttenbach, "Gamebug software," .