Fair and Efficient Secure Multiparty Computation with Reputation Systems*

Gilad Asharov, Yehuda Lindell, and Hila Zarosim**

Dept. of Computer Science, Bar-Ilan University, Israel
{asharog,zarosih}@cs.biu.ac.il, lindell@biu.ac.il

Abstract. A reputation system for a set of entities is essentially a list of scores that provides a measure of the reliability of each entity in the set. The score given to an entity can be interpreted (and in the reputation system literature it often is [12]) as the probability that an entity will behave honestly. In this paper, we ask whether or not it is possible to utilize reputation systems for carrying out secure multiparty computation. We provide formal definitions of secure computation in this setting, and carry out a theoretical study of feasibility. We present almost tight results showing when it is and is not possible to achieve *fair* secure computation in our model. We suggest applications for our model in settings where some information about the honesty of other parties is given. This can be preferable to the current situation where either an honest majority is arbitrarily assumed, or a protocol that is secure for a dishonest majority is used and the efficiency and security guarantees (including fairness) of an honest majority are not obtained.

Keywords: secure multiparty computation, reputation systems, new models

1 Introduction

1.1 Background

In the setting of secure multiparty computation, a set of mutually distrustful parties $P_1, \ldots, P_m$ wish to compute a function of their inputs in the presence of adversarial behavior. The security requirements of such a computation are that nothing beyond the output should be learned (privacy), the output received must be correctly computed (correctness), the parties must choose their inputs independently of each other (independence of inputs), and either no parties

 This research was supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 239868.

** Hila Zarosim is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

receive output or all parties receive output (fairness). The formal definition of security requires that the result of a secure protocol be like the outcome of an ideal execution where an incorruptible trusted party is used to compute the function for all the parties. We remark that if there is no honest majority, then it is impossible to achieve fairness in general [8].

Under the assumption that the majority of the parties are honest, there exist protocols with full security [3, 7, 15]. However, the security of known protocols totally collapses when this assumption does not hold; in particular, the adversary can learn the inputs of the honest parties. Based on this, it may seem prudent to use protocols that guarantee security *except for fairness* when any number of parties are corrupted [15]. Unfortunately, all known protocols of this type have the property that *just one corrupted party* can prevent the parties from terminating successfully, and can even breach fairness. Moreover, it is known that there exist no protocols that simultaneously achieve full security for the case of honest majority, and security-with-abort (i.e., without fairness) when there is no honest majority [18]. This leads to the following unfortunate situation: the parties need to make a decision in advance whether to run a protocol that is secure as long as there is an honest majority and thereby risk losing privacy if they are wrong, or to run a protocol that is secure for any number of corruptions and thereby give up on any hope of obtaining fairness. To make things worse, *this decision is essentially made with no concrete information.*

Reputation systems. A reputation system is a system whose aim is to predict agents' behavior in future transactions. Such systems evaluate the data about agents' previous transactions and estimate the probability that an agent will behave honestly or dishonestly in future transactions [12]. Reputation systems are very popular today in the electronic commerce market and in peer to peer systems [2]. They are used in these contexts to choose which vendors are trustworthy, to determine the level of service obtained by a peer, and more. There is considerable work on how to construct reliable reputation systems, maintain them and so on. Such systems provide us with information regarding the honesty of parties, and therefore could be utilized.

Reputation systems and secure computation. In this paper, we study the use of reputation systems in order to carry out secure multiparty computation. We consider a model where all parties are given a reputation vector $(r_1, \ldots, r_m)$ with the interpretation that the probability that party P_i is honest is r_i. Another possible interpretation of this model is that there exists an adversary who attempts to corrupt as many parties as possible. Then, r_i is the probability that party P_i remains uncorrupted, and can depend on the security measures employed by party P_i. The main question that we ask is:

> *Can reputation systems be utilized in order to achieve fair and efficient secure multiparty computation?*

This model differs from the standard model of secure computation since all parties are given information about the honesty of the other parties and the level at which they can be trusted. Thus, there is hope that this can be used

to achieve more than is possible in the standard model. For example, it may be possible to use protocols that require an honest majority (that are more efficient than those for a dishonest majority, and in addition also guarantee fairness), without just arbitrarily hoping that a majority of the parties are honest. It is important that this actually models a more general setting than just that of reputation systems; see below.

1.2 Our Results

Our main contributions are as follows. First, we suggest this novel model for secure computation and provide formal definitions of security in this model. Next, we study the problem of secure computation with reputation systems from a theoretical perspective. Specifically, we ask under what conditions on the reputation vector it is possible to achieve *fair* secure multiparty computation. We stress that our focus is on fairness since without this requirement one can just ignore the reputation system entirely and run a protocol like [15] that assumes no honest majority and guarantees security without fairness. We present both feasibility and impossibility results for this setting.

Regarding feasibility, we provide a criterion for when the reputations are such that there exists a subset of parties for which a majority are honest, except with negligible probability. Thus, when a reputation vector fulfills the criterion, it is possible to have this subset run a secure protocol that assumes an honest majority. Using the protocol of [10], this subset can be used to run the protocol for many other parties who just provide input and receive output (and it does not matter how many of these other parties are corrupted). Regarding impossibility, we present another criterion on the reputations and show that when this criterion is fulfilled it is impossible to securely toss a coin. This is proven by showing a reduction to the case of two-party coin tossing in the standard model of secure computation, for which the impossibility of fair coin tossing is well known [8]. Interestingly, we show that in the case of *constant* reputation values (that do not depend on the security parameter), our characterization is tight. That is, we prove the following very informally stated theorem:

Theorem 1.1 (Feasibility Characterization – Informal Statement) *Let* Rep *be a reputation system. Then, there exist protocols for securely computing every family of functionalities* $\mathcal{F}$ *with complete fairness with respect to* Rep *if and only if the number of parties with reputation greater than* $1/2$ *is superlogarithmic in the security parameter* n*.*

As we have mentioned, the positive result is obtained by showing that when the condition on the number of parties with reputation greater than $1/2$ is fulfilled then there exists a subset of parties within which there is an honest majority, except with negligible probability. Thus, standard protocols for secure computation with fairness can be run by this subset.

The main question that this leaves is whether or not it is possible to use a reputation system to achieve fairness in a different and more "interesting" way

than just finding a subset for within which there is an honest majority. We show that in fact it is *impossible* to utilize reputations systems in any other way, and so our upper bound is almost tight.[1]

We also show how it is possible to use our feasibility result given a concrete reputation system; This is not immediate since our theoretical feasibility result is asymptotic also in the number of parties and requires finding a subset of parties whose reputation values fulfill a special property.

Reputation systems with correlations. The aforementioned basic model implicitly assumes independence between parties, since each party P_i is corrupted with probability r_i as given in the reputation vector. This therefore does not model the case that P_i and P_j are both corrupted if and only if some P_k is corrupted. We therefore also study the more general setting where the probabilities that parties are corrupted may be correlated. In this setting, we show that as long as the correlations are "limited" in the sense that each party is dependent on only ℓ other parties, then an honest majority exists (except with negligible probability) if the expected number of honest parties is "large enough". We formally define what it means for correlations to be limited to ℓ, and give a criterion on the required expected number of honest parties. We prove this using martingales.

We remark that although this extension allows a more general type of reputation system, real-world reputation systems work by providing a vector stating the individual probabilities that every party is corrupted. Thus, we view the basic model as our main model.

Covert security. We observe that the model of security in the presence of covert adversaries [1], where the guarantee is that any cheating is detected by honest parties, is particularly suited to our setting where there is an existing reputation system. This is due to the fact that any cheating will go immediately punished by reporting such a cheating to the reputation system manager. In addition, we observe that it is possible to use the protocol of [9] that is only twice as expensive as a semi-honest information-theoretic protocol, and provides a deterrent of 1/4 (meaning that any cheating is detected with probability at least 1/4). Such protocols have been proven to be highly efficient.

Applications to other settings. Our basic model for secure computation with reputations actually relates to any setting where additional information about the honesty of the parties is known. Two examples of such settings are as follows. First, consider an environment with an access control scheme where there is non-negligible probability of impersonation. Expressing this probability of cheating as a reputation system and using our protocols, it is possible to neutralize the threat from the impersonators. A second example relates to a set of servers where intrusion detection tools provide an indication as to whether

[1] We remark that in the general case that the parties' reputations may depend on the security parameter, our results are *not* completely tight; in the full version of this paper, we present a concrete example of reputation values for which neither our feasibility result nor impossibility result holds.

or not a given server has been compromised. Rather than naively assuming that a majority of the servers have not been compromised, it is possible to use the indicators of the intrusion detection system within our protocol in order to obtain a more robust solution. This setting fits in very nicely with numerous recent projects that offer a secure computation service, where a set of servers carry out the computation and security is guaranteed as long as a majority of them are not compromised [4, 5].

2 Definitions – The Basic Model

2.1 Reputation Vectors and Secure Computation

Let f be an m-ary functionality and let π be an m-party protocol for computing f. A reputation vector $\boldsymbol{r}$ for the protocol π is a vector in $\mathbb{R}^m$ such that for every i, the value r_i indicates the probability that party P_i is honest in an execution of π. We assume that $\boldsymbol{r}$ is public information, and is obtained from an external authority that handles the reputation system. Our goal in this work is to study the advantages that such public information can provide in constructing secure protocols. We remark that reporting malicious behavior of individuals and maintaining the reputation system is out of this scope of this work. A huge amount of work deals with how to adapt the reputation of an individual in case it has been corrupted. Incorporating cryptography to this task and proving the system that indeed the entity behaves inappropriately, seems as an appealing future direction.

In the standard setting of secure computation, we are given a fixed m-ary function and our goal is to construct a secure m-party protocol π for computing f. Thus the functionality to be computed is fixed and hence its arity is fixed as well. However, in this paper we wish to work asymptotically in the number of parties as well, and this makes things more complicated. The reason that we work in this way is so that we can reason about the probability that some subset of parties of a given size is corrupted. In order to see this, consider a protocol that is secure as long as the majority of the parties are honest. Then, consider the case that all parties are honest with probability 3/4 (and otherwise they are corrupted). Clearly, for a sufficiently large number of parties it is possible to apply the Chernoff bound in order to argue that the probability that there is no honest majority is negligible. However, this is only possible when we consider an asymptotic analysis over the number of parties. We stress that just like the use of a security parameter, in a real instantiation of a protocol one would set a concrete allowed error probability (e.g., 2^{-40}) and verify that for the given real number of parties and their reputation vector, the protocol error is below this allowed probability.

Toward this end, we consider a family of functionalities, each with a different arity, rather than considering a fixed functionality. We require the existence of a polynomial-time process that is given the requested arity m and security parameter n and outputs a circuit $C_{n,m}$ for computing the functionality f^m; this suits the natural case that f computes the same function for each m and the

only difference is the number of inputs (e.g., statistics like median, majority and so on). Formally,

Definition 2.1 *Let $\mathcal{F} = \{f^m\}_{m\in\mathbb{N}}$ be an infinite family of functionalities, where f^m is an m-ary functionality. We say that $\mathcal{F}$ is a* PPT family of functionalities *if there exists a polynomial $p(\cdot)$ and a machine M that on input n and m outputs a circuit $C_{n,m}$ in time at most $p(n+m)$ such that for every $x_1, \ldots, x_m$, it holds that $C_{n,m}(x_1, \ldots, x_m) = f^{(m)}(1^n, x_1, \ldots, x_m)$.*

We define a family of protocols $\Pi(m, n)$ in the same way, and say that it is polynomial time if there exists a polynomial $p(\cdot)$ such that the running time of all parties is bounded by $p(m+n)$. We will consider the case where the number of parties $m = m(n)$ is bounded by a polynomial in the security parameter n. This makes sense since any given party cannot run more than $\text{poly}(n)$ in any case, and so if the number of parties $m(n)$ is superpolynomial in n, then it will not be possible to even send a single message to all other parties.

Summary. We consider secure computation with $m = m(n)$ parties, where $m : \mathbb{N} \to \mathbb{N}$ is bounded by a polynomial in n.[2] The parties run a protocol $\Pi(m, n)$, which is an m-party protocol with security parameter n, that securely computes the functionality f^m in the class $\mathcal{F} = \{f^{m(n)}\}_{n\in\mathbb{N}}$. Finally, the parties have for auxiliary input a reputation vector $\boldsymbol{r}^m$ such that for every $i \in [m]$, the probability that party P_i is corrupted is r_i^m. As we will see, we will require that for all large enough values of n (which also determines $m = m(n)$), the protocol $\Pi(m, n)$ securely computes f^m with respect to the reputation vector $\boldsymbol{r}^m$. Thus, we also need to consider a family of reputation vectors, one for each value of m; we denote the family of reputation vectors for every n by $\mathsf{Rep} = \{\boldsymbol{r}^{m(n)}\}_{n\in\mathbb{N}}$.

2.2 Security with Respect to a Reputation Vector

We assume that the reader is familiar with the standard definition of security for secure computation (see [14, 6]). We modify the definition to allow for a varying number of parties, that is, $m(n)$ for a given function $m : \mathbb{N} \to \mathbb{N}$.

Definition 2.2 *Let $m : \mathbb{N} \to \mathbb{N}$ be a function. We say that the protocol Π $t(\cdot)$-*securely computes *the functionality $\mathcal{F} = \{f^{m(n)}\}_{n\in\mathbb{N}}$ with respect to $m(\cdot)$, if for every* PPT *adversary $\mathcal{A}$, there exists a* PPT *simulator $\mathcal{S}$, such that for every* PPT *distinguisher D, there exists a negligible function $\mu(\cdot)$ such that for every $n \in \mathbb{N}$, every $I \subseteq [m(n)]$ with $|I| \leq t(m(n))$, every $\boldsymbol{x} \in (\{0,1\}^*)^{m(n)}$ and $z \in \{0,1\}^*$, it holds that:*

$$\left|\Pr\left[D\left(\text{IDEAL}_{\mathcal{F},\mathcal{S}(z),I}(n, m, \boldsymbol{x})\right) = 1\right] - \Pr\left[D\left(\text{REAL}_{\Pi,\mathcal{A}(z),I}(n, m, \boldsymbol{x})\right) = 1\right]\right| \leq \mu(n).$$

[2] We remark that the naive approach of taking m to be a parameter that is independent of n does not work, since security would also need to hold when m is superpolynomial in n. However, in such a case, one cannot rely on cryptographic hardness. In addition, bounding m by a polynomial in n is natural in the sense that parties cannot run in time that is superpolynomial in n in any case.

The protocol of GMW [15, 14] satisfies this definition, and is secure even when the number of parties m is a function of n, as long as it is *polynomial*. Formally, let $\Pi(m,n)$ denote the GMW protocol with the following change. Let $\mathcal{F} = \{f^{m(n)}\}_{n\in\mathbb{N}}$ be a functionality. Then, upon input $1^n, 1^m$, each party runs the polynomial-time process to obtain the circuit $C_{n,m}$ for computing f^m. They then proceed to run the GMW protocol with m parties on this circuit. We are interested here in the version of GMW that assumes an honest majority and guarantees fairness. Thus, we have:

Fact 2.3 *Let $\mathcal{F} = \{f^{m(n)}\}_{n\in\mathbb{N}}$ be a functionality and let Π denote the GMW protocol as described above for $\mathcal{F}$. Then, for every polynomial $m(\cdot) : \mathbb{N} \to \mathbb{N}$, the protocol $\Pi(m,n)$ $\frac{m(n)}{2}$-securely computes $\mathcal{F}$ with respect to $m(n)$.*

Having defined security with respect to a varying number of parties, and thus actually being asymptotic also in the number of parties, we proceed to include the reputation system as well. The definition is the same except that instead of quantifying over *all possible subsets of corrupted parties of a certain size*, the set of corrupted parties is chosen probabilistically according to the given reputation vector $\boldsymbol{r}^m = (r_1^m, \ldots, r_m^m)$. We denote by $I \leftarrow r^m$ the subset $I \subseteq [m]$ of parties chosen probabilistically where every $i \in I$ with probability $1 - r_i^m$ (independently of all $j \neq i$). We note that the output of IDEAL and REAL includes $1^n, 1^m, \boldsymbol{x}, z$ and I. Thus the probabilistic choice of I is given to the distinguisher.

Definition 2.4 (Security with Respect to $(m(\cdot), \mathsf{Rep})$) *Let $m(\cdot)$, Rep, $\mathcal{F}$ and Π be as above. We say that Π securely computes $\mathcal{F}$ with respect to $(m(\cdot), \mathsf{Rep})$, if for every PPT adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$, such that for every PPT distinguisher D, there exists a negligible function $\mu(\cdot)$ such that for every $n \in \mathbb{N}$, every $\boldsymbol{x} \in (\{0,1\}^*)^{m(n)}$ and $z \in \{0,1\}^*$, it holds that:*

$$\Big| \Pr_{I \leftarrow \boldsymbol{r}^{m(n)}} \left[D\left(\text{IDEAL}_{\mathcal{F},\mathcal{S}(z),I}\left(n, m(n), \boldsymbol{x}\right)\right) = 1\right] - \Pr_{I \leftarrow \boldsymbol{r}^{m(n)}} \left[D\left(\text{REAL}_{\Pi,\mathcal{A}(z),I}\left(n, m(n), \boldsymbol{x}\right)\right) = 1\right] \Big| \leq \mu(n)$$

Observe that a protocol that is secure with respect to a reputation vector is allowed to always fail for a certain subset I of corrupted parties, if that specific corruption subset is only obtained with negligible probability with the reputation vectors in Rep.

3 A Theoretical Study

In this section we explore our model, and ask under what conditions on the reputation vector, security can be obtained. We first observe that when an honest majority (over all or just a subset of parties) can be guaranteed except with negligible probability, then it is possible to run protocols that are secure with an honest majority like [15, 14] and [21]. We then present a simple condition on a family of reputation vectors that determines whether or not an honest majority

on a subset exists. We also present a condition on the reputation vectors for which it is impossible to securely compute the coin tossing functionality. This is shown by reduction to the impossibility of computing two-party coin tossing with fairness [8]. Finally, we show that when the reputations are *constant*, then the above conditions are complementary. In the full version we give an example of a reputation vector with probabilities that depend on n for which neither of our conditions apply.

3.1 Feasibility

Reputation Vectors and Honest Majority We begin by presenting a simple property for evaluating whether or not a family of reputation vectors guarantees an honest majority, except with negligible probability. It is clear that if all parties have reputation $r_i > \frac{1}{2} + \epsilon$ for some constant ϵ, then there will be an honest majority except with probability that is negligible in the number of parties; this can be seen by applying the Chernoff bound. Likewise, if at least two thirds of the parties have reputation $r_i > \frac{3}{4} + \epsilon$, then a similar calculation will yield an honest majority except with negligible probability. However, these calculations require a large subset of parties to have high reputation, and the use of Chernoff requires that we use the same probability for a large set. Thus, this does not address the case that 1/4 of the parties have very high reputation (almost 1), another half have reputation 1/2, and the remaining 1/4 have low reputation. In order to consider this type of case, we use the Hoeffding Inequality [19]. This enables us to relate to the overall *sum* (or equivalently, *average*) of the reputations of all parties. Using this inequality, we obtain a very simple condition on reputation vectors. Namely, given a family $\mathsf{Rep} = \{\boldsymbol{r}^{m(n)}\}_{n\in\mathbb{N}}$ and a polynomial $m = m(n)$, we simply require that for all sufficiently large n's, the average of the reputations is greater than: $1/2 + \omega\left(\sqrt{\frac{\log m}{m}}\right)$, or, equivalently, that the expected number of honest parties is greater than: $m/2 + \omega\left(\sqrt{m \cdot \log m}\right)$.

Before proceeding to the formal proof, we first state the Hoeffding Inequality [19] (see also [13, Sec. 1.2]). In our specific case, all of the random variables have values between 0 and 1, and we therefore write a simplified inequality for this case.

Lemma 3.1 (The Hoeffding Inequality) *Let $X_1, \ldots, X_m$ be m independent random variables, each ranging over the (real) interval $[0,1]$, and let $\mu = \frac{1}{m} \cdot \mathrm{E}[\sum_{i=1}^{m} X_i]$ denote the expected value of the mean of these variables. Then, for every $\epsilon > 0$, $\Pr\left[\left|\frac{\sum_{i=1}^{m} X_i}{m} - \mu\right| \geq \epsilon\right] \leq 2 \cdot e^{-2\epsilon^2 \cdot m}$.*

Claim 3.2 *Let $m : \mathbb{N} \to \mathbb{N}$ be such that $O(\log m(n)) = O(\log n)$, let $\mathsf{Rep} = \{\boldsymbol{r}^{m(n)}\}_{n\in\mathbb{N}}$ be a family of reputation vectors and let $m = m(n)$. If it holds that*

$$\sum_{i=1}^{m} r_i^m > \left\lfloor \frac{m}{2} \right\rfloor + \omega\left(\sqrt{m \cdot \log m}\right),$$

then there exists a negligible function $\mu(n)$ such that for every n,

$$\Pr_{I \leftarrow \boldsymbol{r}^m}\left[|I| \geq \left\lfloor \frac{m}{2} \right\rfloor\right] < \mu(n) \ .$$

Proof: Fix n and let $m = m(n)$. For every $i \in [m]$, let X_i be a random variable that equals 1 when party P_i is honest, and 0 when it is corrupted. Thus, $\Pr[X_i = 1] = r_i$. Let $\bar{X} = \frac{\sum_{i=1}^m X_i}{m}$. Using linearity of expectations, we have that $\mathrm{E}[\bar{X}] = \frac{1}{m}\sum_{i=1}^m r_i$.

There is an honest majority when $|I| < \frac{m}{2}$; equivalently, when $\sum_{i=1}^m X_i \geq \lfloor m/2 \rfloor + 1$. Let $\Delta = (\sum_{i=1}^m r_i) - \lfloor m/2 \rfloor = m \cdot \mathrm{E}[\bar{X}] - \lfloor m/2 \rfloor$. By the Hoeffding inequality:

$$\begin{aligned}
&\Pr\left[\sum_{i=1}^m X_i \leq \left\lfloor \frac{m}{2} \right\rfloor\right] = \Pr\left[\sum_{i=1}^m X_i - m \cdot \mathrm{E}\left[\bar{X}\right] \leq \left\lfloor \frac{m}{2} \right\rfloor - m \cdot \mathrm{E}\left[\bar{X}\right]\right] \\
&= \Pr\left[\sum_{i=1}^m X_i - m \cdot \mathrm{E}\left[\bar{X}\right] \leq -\Delta\right] = \Pr\left[\sum_{i=1}^m X_i - m \cdot \mathrm{E}\left[\bar{X}\right] \leq -m \cdot \frac{\Delta}{m}\right] \\
&= \Pr\left[\frac{\sum_{i=1}^m X_i}{m} - \mathrm{E}\left[\bar{X}\right] \leq -\frac{\Delta}{m}\right] \leq 2e^{-\frac{2\Delta^2}{m}} \ .
\end{aligned}$$

The above holds for all n and $m = m(n)$. Asymptotically, by the assumption in the claim, $\Delta = \omega(\sqrt{m \cdot \log m})$ and thus $\frac{\Delta^2}{m} = \omega(\log m)$. Hence we have that, $\Pr\left[|I| \geq \left\lfloor \frac{m}{2} \right\rfloor\right] = \Pr\left[\sum_{i=1}^m X_i \leq \left\lfloor \frac{m}{2} \right\rfloor\right] \leq 2e^{-\frac{2\Delta^2}{m}} < 2e^{-\omega(\log m)}$ which is negligible in m. Since $m(\cdot)$ is a function such that $O(\log m(n)) = O(\log n)$, it holds that $e^{-\omega(\log m(n))}$ is a function that is negligible in n, as required. ■

Intuitively, in order to use the above for secure computation, all the parties need to do is to run a protocol that is secure with an honest majority (like GMW [15]). Since there is guaranteed to be an honest majority except with negligible probability, then this is fine. We stress, however, that for this to work we need to use Fact 2.3 since here we refer to the version of GMW for which the number of parties m is a *parameter*, and is not fixed. We therefore conclude:

Theorem 3.3 *Let $\mathcal{F}$ be as above, and let $\Pi = \{\Pi(m,n)\}$ be the GMW protocol of Fact 2.3. Let $m(\cdot)$ be a function such that $O(\log m(n)) = O(\log n)$, let $m = m(n)$ and let* Rep *be as above. If*

$$\sum_{i=1}^m r_i^m > \left\lfloor \frac{m}{2} \right\rfloor + \omega(\sqrt{m \log m}) \ ,$$

then Π securely computes $\mathcal{F}$ with respect to $(m(\cdot), \mathsf{Rep})$.

The proof of this is immediate; if there is an honest majority then the real and ideal executions are indistinguishable, and there is an honest majority except with negligible probability.

Subset Honest Majority In order to achieve secure computation with complete fairness, it suffices to have a subset of parties for which there is guaranteed to be an honest majority except with negligible probability [10]. This works by having the subset carry out the actual computation for all other parties. Specifically, all the parties send shares of their inputs to the subset, who compute shares of the output and return them. In more detail, the protocol of [10] works as follows. Let $T \subseteq [m(n)]$ be the subset of parties that carry out the actual computation; these are called the *servers*. All the parties distribute their inputs to the set of servers T using VSS (verifiable secret sharing) with threshold $|T|/2+1$. The servers then compute shares of the outputs by computing the circuit gate by gate. At the output phase, the servers send the appropriate shares of the outputs to each party, who then reconstructs the output. See [10] for details, and for a proof that the protocol is secure as long as a majority of the parties in T are honest. Thus, as long as there exists a subset of parties $T \subseteq [m(n)]$ with honest majority except with negligible probability, there exists a protocol for this $m(\cdot)$ and family of reputation vectors. Thus, we have:

Claim 3.4 *Let $\mathcal{F}$, $m(\cdot)$ and Rep be as above. If there exists a negligible function $\mu(\cdot)$, such that for every n there exists a subset $T_n \subset [m(n)]$ for which $\Pr_{I \leftarrow \boldsymbol{r}^{m(n)}}\left[|T_n \cap I| \leq \frac{|T_n|}{2}\right] \leq \mu(n)$, then there exists a (non-uniform) protocol Π that securely computes $\mathcal{F}$ with respect to $(m(\cdot), \mathsf{Rep})$.*

The proof of this claim is the same as the proof of Theorem 3.3: if there is a subset with an honest majority then the security of [10] holds, and the probability that there is not an honest majority is negligible. There is one subtle point here, which is that the protocol as described is *non-uniform* since the subset T_n may be different for every n. Nevertheless, as we will see below, our criteria for the existence of such a subset is such that an appropriate subset T_n can always be efficiently computed from the reputation vector $\boldsymbol{r}^{m(n)}$ (assuming its existence).

Criteria on the reputation vector. Our next goal is to analyze when a family of reputation vectors guarantees a subset of parties with an honest majority, except with negligible probability. We first present the technical criteria, and then explain its significance below.

Claim 3.5 *Let $m(\cdot)$, and Rep be as above. For every n and subset $T_n \subseteq [m(n)]$, let $\Delta_{T_n} \stackrel{\text{def}}{=} \sum_{i \in T_n} \boldsymbol{r}_i^{m(n)} - \frac{|T_n|}{2}$. If there exists a series of subsets $\{T_n\}_{n \in \mathbb{N}}$ (each $T_n \subseteq [m(n)]$) such that $\frac{(\Delta_{T_n})^2}{|T_n|} = \omega(\log n)$, then there exists a negligible function $\mu(\cdot)$ such that for every n, $\Pr_{I \leftarrow \boldsymbol{r}^{m(n)}}\left[|I \cap T_n| > \frac{|T_n|}{2}\right] \leq \mu(n)$.*

Proof: The proof of this claim is very similar to that of Claim 3.2, and uses the Hoeffding inequality. Let $\{T_n\}_{n \in \mathbb{N}}$ be a series of subsets as in the claim. Fix n and $T = T_n$. Then, for every $i \in T$, let X_i be a random variable that equals 1

when the party is honest and 0 when it is corrupted. An identical calculation to that carried out in the proof of Claim 3.2 yields that for every n,

$$\Pr_{I \leftarrow \boldsymbol{r}^{m(n)}}\left[|I \cap T_n| > \frac{|T_n|}{2}\right] = \Pr\left[\sum_{i \in T_n} X_i \leq \frac{|T_n|}{2}\right] \leq 2e^{-\frac{2(\Delta_{T_n})^2}{|T_n|}}. \quad (1)$$

Since $\frac{(\Delta_{T_n})^2}{|T_n|} = \omega(\log n)$, we conclude that $\Pr_{I \leftarrow \boldsymbol{r}^{m(n)}}\left[|I \cap T_n| > \frac{|T_n|}{2}\right] \leq 2e^{-\omega(\log n)}$ which is negligible in n, as required. ■

Combining Claim 3.5 with Claim 3.4 we conclude that:

Corollary 3.6 *Let $\mathcal{F}$, $m(\cdot)$ and* Rep *be as above. For every n and subset $T_n \subseteq [m(n)]$, let $\Delta_{T_n} \stackrel{\text{def}}{=} \sum_{i \in T_n} \boldsymbol{r}_i^{m(n)} - \frac{|T_n|}{2}$. If there exists a series of subsets $\{T_n\}_{n \in \mathbb{N}}$ (each $T_n \subseteq [m(n)]$) such that $\frac{(\Delta_{T_n})^2}{|T_n|} = \omega(\log n)$, then there exists a (non-uniform) protocol Π that securely computes $\mathcal{F}$ with respect to $(m(\cdot), \mathsf{Rep})$.*

Efficiently finding the subset T_n. The non-uniformity of the protocol is due to the fact that in general, the subset T_n may not be efficiently computable from the reputation vector. Nevertheless, we show now that assuming the existence of a subset T_n fulfilling the condition, it is easy to find a subset T'_n (not necessarily equal to T_n) that also fulfills the condition.

In order to see this, first note that for any size t, the largest value of Δ (over all subsets $T_n \subseteq [m(n)]$ of size t) is obtained by taking the t indices i for which r_i is largest. This follows since $\Delta_{T_n} = (\sum_{i \in T_n} r_i) - \frac{|T_n|}{2}$ and so replacing an r_i in the sum with a larger r_j always gives a larger Δ_{T_n}. This gives the following algorithm for finding an appropriate subset:

1. Given $\boldsymbol{r}^m$, sort the values in decreasing order; let $r_{i_1}, \ldots, r_{i_m}$ be the sorted values.
2. For every $j = 1, \ldots, m$, compute $\Delta_j = \left(\sum_{k=1}^{j} r_{i_k}\right) - \frac{j}{2}$.
3. Let j be the index for which $\frac{(\Delta_j)^2}{j}$ is maximum over all j's. Then, output the subset $T = \{i_1, \ldots, i_j\}$.

In order to see that this fulfills the requirement, observe that by the above observation, the maximum value of Δ_{T_n} for *all possible subsets* T_n is one of the values of $\Delta_1, \ldots, \Delta_m$. Therefore, if there exists a subset that fulfills the requirement, the subset output by the algorithm also fulfills the requirement.

A Protocol for the Concrete Setting. Our protocols above are proven secure under the *assumption* that there exists a subset T_n fulfilling the required *asymptotic* property. However, concretely, how can a set of parties know that there exists such a subset, and which subset to take? This turns out to be very easy since the Hoeffding inequality is exact, and not asymptotic. Thus, for a given error parameter δ (e.g., $\delta = 2^{-40}$ or $\delta = 2^{-80}$) and subset T_n, it is possible

to simply compute Δ_{T_n} and then check if $2e^{-\frac{2(\Delta_{T_n})^2}{|T_n|}} < \delta$ (this bound is obtained similarly to the bound in the proof of Claim 3.2; see Eq. (1) for more details). By the algorithm given above for finding the subset, it is possible to efficiently find an appropriate T_n with an error below the allowed bound, if one exists. (If one does not exist, then the parties know that they cannot run the protocol.) We remark that for efficiency, it is best to take the smallest subset that gives a value below the allowed error parameter, since this means that the protocol run by the parties has less participants, and so is more efficient.

Inaccurate reputation system. Sometimes, the reputation system may be inaccurate, where the true reputation value of the party is ϵ-far from its public value. This error may arise in both directions, that is, sometimes the public reputation might be lower than the true one, and sometimes it might be higher. However, it is easy to generalize our results to deal with this case as well, while taking the reputation as the minimum guaranteed value and considering the worst case scenario.

3.2 Impossibility

We now turn to study under what conditions on the family of reputation vectors it is not possible to achieve (general) secure computation. We stress that we focus on the question of fairness here since one can always ignore the reputation vector and run a general protocol for secure computation *with abort* that is resilient to any number of corrupted parties. We therefore consider the problem of coin tossing, since it is impossible to fairly toss a coin without an honest majority [8] (or, more accurately, with only two parties).

Let $m(\cdot)$ be a function and let $\mathsf{Rep} = \{\boldsymbol{r}^{m(n)}\}_{n\in\mathbb{N}}$ be a family of reputation vectors. For every n, we denote by $\mathsf{H}_n^{1/2}$ the set of all indices i of parties P_i such that $\frac{1}{2} < r_i^{m(n)} < 1$. Recall that $m(\cdot)$ denotes the number of parties and so the size of the set $\mathsf{H}_n^{1/2}$ is bounded by $m(n)$. We denote by $\mathcal{F} = \{f_{CT}^{m(n)}\}_{n\in\mathbb{N}}$ the coin-tossing functionality: $f_{CT}^{m(n)}(1^n,\ldots,1^n) = (U_1,\ldots,U_1)$, where U_1 denotes a uniform random bit; i.e., the output of the function is the same random bit for all parties.

The idea behind our proof of impossibility is as follows. Consider first for simplicity the case that all the reputations are at most $1/2$, and thus $\mathsf{H}_n^{1/2}$ is empty. This means that the expected number of corrupted parties is at least half and thus, intuitively, any protocol that is secure with respect to such a reputation vector must be secure in the presence of a dishonest majority. We show that this implies the existence of a two party protocol for fair coin tossing. We also prove impossibility when $\mathsf{H}_n^{1/2}$ is not empty but the probability of all parties in $\mathsf{H}_n^{1/2}$ being corrupted is non-negligible. In this case, we show that since security must hold even when all parties in $\mathsf{H}_n^{1/2}$ are corrupted, we can reduce to fair coin tossing even here.

Theorem 3.7 *Let $m(\cdot)$ be polynomially bounded, and let Rep be a family of reputation vectors. If there exists a polynomial $p(\cdot)$ such that for infinitely many*

n's it holds that the probability that all *parties in* $\mathsf{H}_n^{1/2}$ *are corrupted is at least* $\frac{1}{p(n)}$*, then there does not exist a protocol* Π *that securely computes the multiparty coin-tossing functionality* $\mathcal{F} = \{f_{CT}^{m(n)}\}_{n\in\mathbb{N}}$ *with respect to* $(m(\cdot), \mathsf{Rep})$*.*

Proof Sketch: Assume the existence of a protocol Π that securely computes the family $\mathcal{F} = \{f_{CT}^{m(n)}\}_{n\in\mathbb{N}}$ with respect to a polynomial $m(\cdot)$ and a family of reputation vectors Rep, and that there exists a polynomial $p(\cdot)$ such that for infinitely many n's, $\Pr_{I\leftarrow \boldsymbol{r}^{m(n)}}\left[\mathsf{H}_n^{1/2} \subseteq I\right] \geq \frac{1}{p(n)}$. We show that this implies the existence of an infinitely-often[3] non-uniform two-party protocol $\pi' = \langle P_0', P_1'\rangle$ for the coin-tossing functionality that is secure in the presence of malicious adversaries, in contradiction to the fact that fair coin tossing cannot be achieved [8].[4]

We start with an informal description of our transformation and we provide an informal explanation of why it works; we then describe in Protocol 3.8 the construction of π', the formal proof appears in the full version of this paper. We begin with the simpler case where Rep is such that for infinitely many n's, $\mathsf{H}_n^{1/2}$ is empty; that is, each party is honest with probability at most $1/2$. We use this to construct a two-party protocol π' in which on security parameter n, the two parties P_0' and P_1' emulate an execution of the $m = m(n)$-party protocol $\Pi(m, n)$ by randomly choosing which of the m parties in $\Pi(m, n)$ is under the control of P_0' and which of the parties in Π is under the control of P_1'. This can be done by tossing m coins, and giving the control of each (virtual) party for which the coin is 0 to P_0', and the control of each (virtual) party for which the coin is 1 to P_1'. The two parties then emulate an execution of the m-party protocol Π for the coin-tossing functionality f_{CT}^m, and determine the resulting bit according to the outputs of the (virtual) parties under their control.

Loosely speaking, we claim that the security of Π implies that this emulation is secure as well. To see this, note that in an execution of π', the m-party protocol Π is invoked when each of the parties of Π is under the control of P_0' with probability $1/2$ and under the control of P_1' with probability $1/2$. Thus, for every adversary controlling one of the parties in π', we expect to have about half of the parties in Π under its control (since each party in Π is under the adversary's control with probability $1/2$). Since Π is secure with respect to a family of reputation vectors Rep such that $\mathsf{H}_n^{1/2}$ is empty (and so, $r_i^m \leq \frac{1}{2}$ for every i), Π is secure when the expected number of the corrupted parties is $\sum_i (1 - r_i^m) \geq \frac{m}{2}$, and thus can handle the number of corruptions in the emulation carried out by the two party protocol π'.

So far, we have ignored two issues in the construction of π'. First, the coin tossing we use to decide which of the parties is controlled by P_0' and which is controlled by P_1' is only secure with abort, and so an adversary controlling P_0' or P_1' might abort before the other party sees the output of the coin tossing. However, we show that in this case the honest party can simply output a random bit and this adds no bias to the output. Intuitively, the reason that this works

[3] This means that the security of the protocol holds for infinitely many n's.

[4] We note that the proof of impossibility of two-party coin tossing of [8] holds also for infinitely-often non-uniform protocols (the proof of [8] holds for every fixed n).

is that if a party aborts before beginning the emulation of Π, then it has no meaningful information and so cannot bias the outcome. Thus, the other party may just output a random bit. Second, after the emulation ends, each of the parties P'_0 and P'_1 should determine their outputs. Recall that each of the two parties has a subset of the m parties in Π under its control and hence at the end of the emulation of Π, each of P'_0 and P'_1 sees a set of outputs (as each party in Π has an output). However, we expect that when the parties play honestly, all parties in Π output the same output. Moreover, even if some of the parties in Π are corrupted, by the security of Π, the output of the honest parties should be the same (except with a negligible probability). Therefore, intuitively it seems that P'_0 (resp. P'_1) can determine its output by considering the set of all outputs of the parties under its control. If all those parties have the same output, then P'_0 outputs the common bit. Since we expect the event of not all parties having the same output happen only with a negligible probability, in this case P'_0 (resp. P'_1) can just output a $\perp$ symbol. However, when trying to formalize this idea, a technical problem arises because the expected number of honest outputs in π' may be larger than the expected number of honest outputs in Π (recall that in π' the expected number of honest parties is $\frac{m}{2}$ whereas in a real execution of Π the expected number of honest parties is $\sum_i r_i^m \leq \frac{m}{2}$). We overcome this by having the honest party in π' not consider the set of *all* outputs of the parties under its control, but rather choose a subset of the outputs that is expected to be of size $\sum_i r_i^m$. To do this, the parties in π' must know the vector $\boldsymbol{r}^m$ and hence the protocol we construct is non-uniform.

So far we only discussed the case that for infinitely many n's, $\mathsf{H}_n^{1/2}$ is empty. Now, assume that this does not hold and $\mathsf{H}_n^{1/2}$ is non-empty. In this case, the construction of π' fails because in the execution simulated by π', each party is corrupted with probability $\frac{1}{2}$ whereas in the real execution of Π, we have parties whose probabilities to be corrupted are strictly less than $\frac{1}{2}$. For example, assume that a certain party P_i in $\Pi(m,n)$ is honest with probability 0.9 (and hence - corrupted with probability 0.1). However, by the way π' is defined, this party will be under the control of the adversary with probability $\frac{1}{2}$. In this case, it might be that π' is not secure even though Π is secure, simply because the party P_i is more likely to be corrupted in π' than in $\Pi(m,n)$. However, we show that if for infinitely many n's, the probability of all parties in $\mathsf{H}_n^{1/2}$ being corrupted is polynomial in n, then Π must remain (infinitely often) secure even conditioned on the event that the parties in $\mathsf{H}_n^{1/2}$ are corrupted. This will imply that we can slightly modify the construction of π' such that one of the parties always controls the parties in $\mathsf{H}_n^{1/2}$, and obtain that even though these parties are more likely to be corrupted when π' simulates Π than in real executions of Π, the simulation carried out by π' still remains secure.

A formal construction of π' is given in Protocol 3.8 and the full proof that π' securely computes the two-party coin-tossing functionality with fairness appears in the full version of this paper. ∎

PROTOCOL 3.8 (Protocol $\pi' = \langle P'_0, P'_1 \rangle$ for two-party coin-tossing)

- **(Non-uniform) auxiliary input:** 1^n, $\boldsymbol{r}^{m(n)}$.
- **The Protocol:**
  1. *Set up subsets for emulation:* Parties P'_0 and P'_1 invoke $m = m(n)$ executions of the f^2_{CT} functionality with security-with-abort in order to obtain m coins; let $\boldsymbol{b} \in \{0,1\}^m$ be the resulting coins. If one of the parties receives $\perp$ (i.e., abort) for output, then it outputs a random bit and halts.
     Otherwise, the parties define $I_0 = \{i \mid b_i = 0\} \cup \mathsf{H}^{1/2}_n$, and $I_1 = [m] \setminus I_0$.
  2. *Emulate Π:* The parties P'_0 and P'_1 emulate an execution $\Pi = \Pi(m(n), n)$ for computing f^m_{CT} where P'_0 controls the parties in I_0 and P'_1 controls the parties in I_1; all parties use input 1^n.
  3. *Determine outputs:*
     (a) Party P'_0 selects a subset $S^0 \subseteq I_0$ of the (virtual) parties under its control as follows. For every $i \in \mathsf{H}^{1/2}_n$, P_i is added to S^0 with probability r^m_i; for every $i \in I_0 \setminus \mathsf{H}^{1/2}_n$, P_i is added to S^0 with probability $2r^m_i$ (note that since $i \notin \mathsf{H}^{1/2}_n$, it holds that $r^m_i \leq \frac{1}{2}$ and hence $2r^m_i$ is a valid probability).
     P'_0 outputs the bit $b \in \{0,1\}$ if *all* the virtual parties in S^0 output b in Π. Otherwise, it outputs $\perp$.
     (b) Party P'_1 selects a subset $S^1 \subseteq I_1$ of the (virtual) parties under its control by adding each P_i (for $i \in I_1$) with probability $2r^m_i$ (as before, $i \notin \mathsf{H}^{1/2}_n$ and hence $2r^m_i$ is a valid probability).
     P'_1 outputs the bit $b \in \{0,1\}$ if *all* the virtual parties in S^1 output b in Π. Otherwise, it outputs $\perp$.

3.3 Tightness of the Feasibility and Impossibility Results

Our feasibility result states that if there exists a series of subsets $\{T_n\}_{n\in\mathbb{N}}$ (each $T_n \subseteq [m(n)]$) such that $\frac{(\Delta_{T_n})^2}{|T_n|} = \omega(\log n)$, then there exists a secure protocol. In contrast, our impossibility result states that if for infinitely many n's, the probability that all parties in $\mathsf{H}^{1/2}_n$ are corrupted is $1/p(n)$, then there exists no protocol. In this section, we clarify the relation between these two results.

Constant reputations. We consider the case that all reputations are constant. This is somewhat tricky to define since the reputation vectors are modeled asymptotically themselves (each vector of length $m(n)$ can have different values and thus can depend on n). We therefore define "constant" by saying that there exists a finite set $\mathcal{R} = \{r_1, \ldots, r_L\}$ such that all reputation values (in all vectors) in Rep are in $\mathcal{R}$, and $1 \notin \mathcal{R}$ (if $1 \in \mathcal{R}$ then this is an uncorruptible trusted party and secure computation is trivial). In this case, we say that Rep has constant reputations. We have the following theorem:

Theorem 3.9 *Let $m(\cdot)$ be a polynomial, and let* Rep *be a family of* constant *reputations. Then, there exist protocols for securely computing every PPT family of functionalities $\mathcal{F}$ with respect to $(m(\cdot), \mathsf{Rep})$, if and only if it holds that $|\mathsf{H}^{1/2}_n| = \omega(\log n)$.*

Proof: The existence of protocols for every family of functionalities $\mathcal{F}$ when $|\mathsf{H}_n^{1/2}| = \omega(\log n)$ can be seen as follows. Let r be the smallest value greater than $1/2$ in $\mathcal{R}$. This implies that all reputations in $\mathsf{H}_n^{1/2}$ for all n are at least r. Thus, $\Delta_{\mathsf{H}_n^{1/2}} = \sum_{i \in \mathsf{H}_n^{1/2}} \left(\boldsymbol{r}_i^{m(n)} - \frac{1}{2} \right) \geq \sum_{i \in \mathsf{H}_n^{1/2}} \left(r - \frac{1}{2} \right) = |\mathsf{H}_n^{1/2}| \cdot \left(r - \frac{1}{2} \right)$. Now, take $T_n = \mathsf{H}_n^{1/2}$ and we have that $\frac{(\Delta_{T_n})^2}{|T_n|} \geq \frac{|T_n|^2 \cdot \left(r - \frac{1}{2} \right)^2}{|T_n|} = |T_n| \cdot \left(r - \frac{1}{2} \right)^2 = \omega(\log n)$, where the last equality holds because $(r - 1/2)^2$ is constant, and by the assumption $|T_n| = |\mathsf{H}_n^{1/2}| = \omega(\log n)$. Thus, by Corollary 3.6, there exists a protocol for every family $\mathcal{F}$.

For the other direction, assume that it is not the case that $|\mathsf{H}_n^{1/2}| = \omega(\log n)$, for every n. This implies that there exists a constant c such that for infinitely many n's, $|\mathsf{H}_n^{1/2}| \leq c \cdot \log n$. Now, let r' be the highest value in $\mathcal{R}$. It follows that for every $i \in \mathsf{H}_n^{1/2}$, $r_i \leq r'$. Thus, for infinitely many n's the probability that all parties in $\mathsf{H}_n^{1/2}$ are corrupted is at least $(1 - r')^{-c \cdot \log n}$. Since $(1 - r')$ is constant, $(1 - r')^{-c \cdot \log n}$ is $1/\mathrm{poly}(n)$. Thus, by Theorem 3.7, there exists no protocol for coin-tossing (and so it does not hold that all functionalities can be securely computed). ∎

We conclude that in the case of constant reputations, our results are *tight*. In the full version we give an example of a family of reputation vectors with non-constant reputations, for which neither our upper bound nor lower bound applies.

4 Secure Computation with Correlated Reputations

Until now, we have considered reputation systems where the probability that each party is corrupted is independent of the probability that all other parties are corrupted. This follows directly from how we define reputation systems (and, indeed, the way these systems are typically defined in other fields). A natural question that arises is what happens when the probabilities that parties are corrupted are *not* independent; that is, when there is a correlation between the probability that some party P_i is corrupted and the probability that some P_j (or some subset of other parties) is corrupted. In this section we take a first step towards exploring the feasibility of fair secure computation with correlated reputation systems.

We begin by extending Definition 2.4 to this more general case. First, observe that a reputation system can no longer be represented as a vector of probabilities, since this inherently assumes independence. (Specifically, no vector can represent a corruption situation where with probability $1/2$ both parties P_1 and P_2 are corrupted and with probability $1/2$ they are both honest.) Thus, we represent a reputation system with $m = m(n)$ parties where $m : \mathbb{N} \to \mathbb{N}$ is bounded by a polynomial in n by a probabilistic polynomial-time sampling machine M that receives the security parameter $n \in \mathbb{N}$ and outputs a set $I \subseteq [m(n)]$, such that P_i is corrupted if $i \in I$. We call (m, M) the reputation system.

The definition of security is the natural extension of Definition 2.4. Specifically, we say that a protocol Π securely computes $\mathcal{F}$ with respect to a reputation

system (m, M) if all is the same as in Definition 2.4 except that the set I of corrupted parties is chosen by running $M(1^n)$.

We remark that unlike the case of *reputation vectors*, it does not suffice to look at the expected number of honest parties here. In order to see this, consider the case of m parties such that with probability $1/100$ all the parties but one are corrupted, and with probability $99/100$ all the parties are honest. According to this, the expected number of honest parties is $1 \cdot 1/100 + 99/100 \cdot m$ which is greater than $0.99m$. Nevertheless, the probability that there is a dishonest majority is $1/100$ which is clearly non-negligible. Thus, in the setting of correlated reputations where there is dependency between parties, a high expected number of honest parties does not imply that there is an honest majority except with negligible probability.

We show that when the "amount of dependence" between the parties is limited, then it is possible to obtain fair secure computation. In a nutshell, we show that if each party is dependent on at most $\ell(m)$ other parties, where $\ell(m)$ is some function, and the expected number of honest parties in a sampling by M is $\frac{m}{2} + \omega(\ell(m) \cdot \sqrt{m \log m})$, then there is an honest majority except with negligible probability. Given this fact, it is possible to run any multiparty computation protocol that is secure with an honest majority. Thus, in contrast to the above example, we conclude that when dependence is limited, it is possible to use the expected number of honest parties in order to bound the probability of a dishonest majority. *This is a direct generalization of Theorem 3.3, where the bound in Theorem 3.3 is obtained by setting* $\ell(m) = 1$. This result is proven by defining a martingale based on the random variables indicating the corruption status of each party, and then applying Azuma's inequality. We also consider a generalization of our model, where a party can be correlated also with all other $m - \ell(m)$ parties, but only to a very small extent. In addition, as with the case of reputation vectors, we also show how to extend this result to the case of a large enough subset with high enough expectation.

Another way to interpret the aforementioned result is as follows. In practical applications the reputation system is usually represented as a vector, although dependency between the parties may exist. Thus, the parties do not have all the information about the honesty of the parties. However, our analysis shows that the vector of reputations alone may still be useful. By linearity of expectation, the expected number of honest parties is the sum of reputations even if those are somehow dependent. Therefore, if this sum is large enough, honest majority is guaranteed except with negligible probability, even if there exists some correlation between the parties. (where the allowed correlation can be obtained from the sum of reputations and the number of parties).

We now provide our definition for "limited correlations" and a formal statement of the main result. The full proof, together with additional results appear in the full version of this paper.

Defining limited dependence. Let $X_1, \ldots, X_m$ be Boolean random variables such that $X_i = 1$ if and only if P_i is honest, or equivalently if and only if $i \notin I$. We begin by defining what it means for two parties to be dependent on each other.

It is important to note that in our context the naive approach of saying that P_i and P_j are dependent if the random variables X_i and X_j are dependent does *not* suffice. In order to see this, assume that there are three parties P_1, P_2 and P_3, such that P_3 is honest if and only if only one of P_1 and P_2 is honest, and that P_1 and P_2 are honest with probability $1/2$ (independently of each other). Clearly, by the standard notion of dependence P_1 and P_2 are independent. However, the honesty or lack thereof of P_1 and P_2 is influenced by P_3; stated differently, the random variables X_1 and X_2 are *not independent* when conditioning on X_3. Since we need to consider the global context of who is honest and who is corrupted, in such a case we should define that P_1 and P_2 *are* correlated.

Based on the above discussion, we define a new notion of dependency that we call *correlation amongst* $\mathcal{P}$, where $\mathcal{P} = \{P_1, \ldots, P_m\}$ is the set of all parties. Intuitively, we say that a pair of parties P_i and P_j are correlated amongst $\mathcal{P}$ if there exists a subset of parties in $\mathcal{P}$ such that the random variables X_i and X_j are not independent when conditioning on the corruption status of the parties in the subset. We stress that P_i and P_j are correlated as soon as the above holds for any subset. We believe that this is quite a natural definition that captures the intuitive meaning of dependence where the probability that a party is corrupted can depend on coalitions amongst other parties and whether or not they are corrupted. Formally:

Definition 4.1 (Correlated Amongst $\mathcal{P}$) *Let (m, M) be a reputation system. We say that parties P_i and P_j are* correlated amongst $\mathcal{P}$ *if there exists a subset $S \subseteq [m]$, and Boolean values $b_i, b_j, \{b_k\}_{k \in S}$ such that*

$$\Pr\Big[X_i = b_i \wedge X_j = b_j \;\Big|\; \{X_k = b_k\}_{k \in S}\Big] \neq \Pr\Big[X_i = b_i \;\Big|\; \{X_k = b_k\}_{k \in S}\Big] \cdot \Pr\Big[X_j = b_j \;\Big|\; \{X_k = b_k\}_{k \in S}\Big] \; .$$

Let $\mathcal{D}(i)$ be the set of all parties P_j for which P_i and P_j are correlated amongst $\mathcal{P}$. Intuitively, we say that a reputation system has an ℓ-limited correlation if for every i it holds that the size of $\mathcal{D}(i)$ is at most ℓ; that is the number of parties with which any party P_i is correlated is at most ℓ.

Definition 4.2 (ℓ-Limited Correlation) *Let (m, M) be a reputation system and $\ell = \ell(m)$. We say that (m, M) has an ℓ-*limited correlation *if for every $i \in [m]$, it holds that $|\mathcal{D}(i)| \leq \ell(m)$.*

An honest majority in ℓ-limited correlated reputation systems. We show that if the expected number of honest parties in an ℓ-limited correlated reputation systems is large enough, as a function of m and ℓ, then an honest majority is guaranteed except with negligible probability. The proof of this fact uses martingales and Azuma's inequality, and appears in the full version of this paper. Recall that X_i is a random variable that equals 1 when P_i is honest; thus $\sum_{i=1}^{m} X_i$ gives the number of honest parties. We show that the probability that this sum is less than $m/2$ is negligible.

Theorem 4.3 *Let $m : \mathbb{N} \to \mathbb{N}$ be a function such that $O(\log m(n)) = O(\log n)$, let $(m(n), M)$ be a family of reputation systems, and let $\ell = \ell(m)$ where $m = m(n)$. If $(m(n), M)$ has an ℓ-limited correlation and*

$$\mathrm{E}\left[\sum_{i=1}^{m} X_i\right] \geq \frac{m}{2} + \omega\left(\ell(m) \cdot \sqrt{m \log m}\right),$$

then there exists a negligible function $\mu(\cdot)$ such that for every n,

$$\Pr\left[\sum_{i=1}^{m} X_i \leq \frac{m}{2}\right] < \mu(n).$$

The full proof and additional results appear in the full version of the paper.

5 Reputation and Covert Security

In the model of secure computation in the presence of covert adversaries [1], the security guarantee is that if a party cheats then it will be detected cheating with some probability ϵ (this probability is called the deterrent). The deterrent parameter ϵ can be tailored depending on the requirements. For $\epsilon = 1/2$, the cost of computing is between 2 and 4 times the cost of protocols that are secure for semi-honest adversaries. Thus, this is much more efficient than security in the presence of malicious adversaries. The model of covert adversaries is *particularly suited to a setting with reputation systems* since if cheating is detected, then an immediate "punishment" can be incurred via a report to the reputation system manager. Thus, the use of protocols that are secure for covert adversaries makes real sense here.

In addition to the above, we observe that great efficiency can be obtained by using the protocol of [9]. This protocol assumes an honest majority and obtains security in the presence of covert adversaries with deterrent $\frac{1}{4}$, at just twice the cost of obtaining information-theoretic security in the semi-honest model. Thus, this protocol is extremely efficient. Combining this with the fact that it is possible to run the protocol on the smallest subset that yields an honest majority except with probability below the allowed error δ (see Section 3.1), we have that large-scale computation involving many thousands of participants can be efficiently computed by taking a much smaller subset to run the protocol of [9].

References

1. Y. Aumann and Y. Lindell. Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries. In *4th TCC*, Springer-Verlag (LNCS 4392), pages 137-156, 2007.
2. M. Babaioff, J. Chuang and M. Feldman. Incentives in Peer-to-Peer Systems. In *Algorithmic Game Theory* (Chapter 23), Cambridge University Press, 2007.

3. M. Ben-Or, S. Goldwasser and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In 20*th STOC,* pages 1–10, 1988.
4. P. Bogetoft, D.L. Christensen, I. Damgård, M. Geisler, T.P. Jakobsen, M. Kroigaard, J.D. Nielsen, J.B. Nielsen, K. Nielsen, J. Pagter, M.I. Schwartzbach and T. Toft. Secure Multiparty Computation Goes Live. In *Financial Cryptography*, pages 325–343, 2009.
5. D. Bogdanov, S. Laur and J. Willemson. Sharemind: A Framework for Fast Privacy-Preserving Computations. In the 13*th ESORICS*, Springer (LNCS 5283), pages 192–206, 2008.
6. R. Canetti. Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
7. D. Chaum, C. Crépeau and I. Damgård. Multi-party Unconditionally Secure Protocols. In 20*th STOC*, pages 11–19, 1988.
8. R. Cleve. Limits on the Security of Coin Flips when Half the Processors are Faulty. In 18*th STOC,* pages 364–369, 1986.
9. I. Damgård, M. Geisler and J.B. Nielsen. From Passive to Covert Security at Low Cost. In the 7*th TCC*, Springer (LNCS 5978), pages 128–145, 2010.
10. I. Damgård and Y. Ishai. Constant-Round Multiparty Computation Using a Black-Box Pseudorandom Generator. In *CRYPTO 2005*, Springer (LNCS 3621), pages: 378–394, 2005.
11. D. Dubhashi and A. Panconesi. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms (1st ed.).* Cambridge University Press, New York, NY, USA.
12. E. Friedman, P. Resnick and R. Sami. Manipulation-Resistant Reputation Systems. In *Algorithmic Game Theory* (Chapter 27), Cambridge University Press, 2007.
13. O. Goldreich. *Foundations of Cryptography: Volume 1 – Basic Tools.* Cambridge University Press, 2001.
14. O. Goldreich. *Foundations of Cryptography: Volume 2 – Basic Applications.* Cambridge University Press, 2004.
15. O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In 19*th STOC,* pages 218–229, 1987. For details see [14].
16. V. Goyal, P. Mohassel and A. Smith. Efficient Two Party and Multi Party Computation Against Covert Adversaries. In *EUROCRYPT 2008*, Springer-Verlag (LNCS 4965), pages 289–306, 2008.
17. S. Halevi, Y. Lindell, and B. Pinkas. Secure Computation on the Web: Computing without Simultaneous Interaction. In *CRYPTO 2011*, Springer (LNCS 6841), pages 132–150, 2011.
18. Y. Ishai, J. Katz, E .Kushilevitz, Y. Lindell and E. Petrank. On Achieving the "Best of Both Worlds" in Secure Multiparty Computation. In *SIAM J. Comput.* 40(1), pages 122–141, 2011.
19. W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. In *Journal of the American Statistical Association*, 58(301):13-30, March 1963.
20. M. Mitzenmacher and E. Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA.
21. T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multi-party Protocols with Honest Majority. In 21*st STOC*, pages 73–85, 1989.