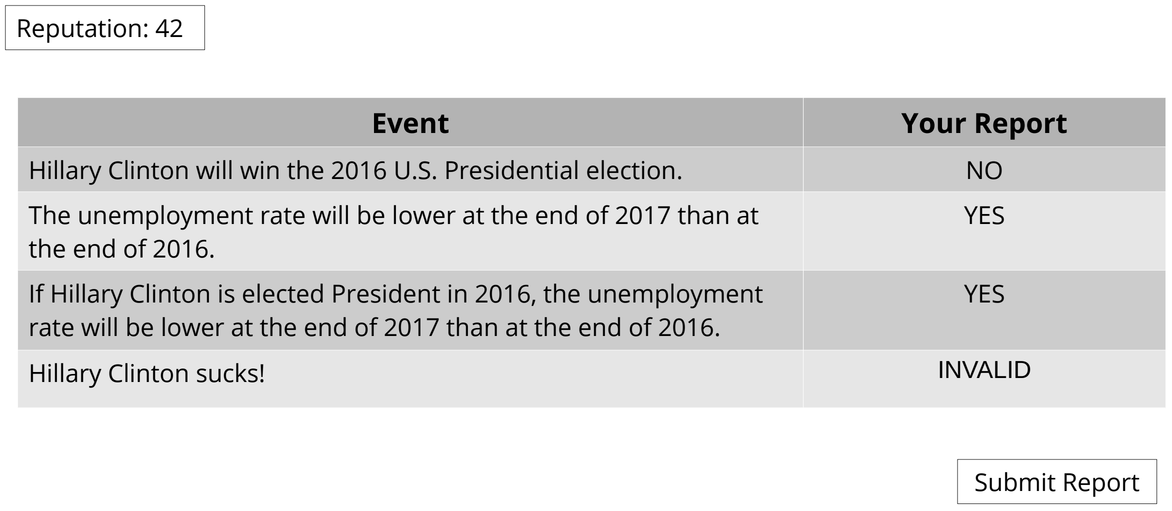

Reputation: 42

Event	Your Report
Hillary Clinton will win the 2016 U.S. Presidential election.	NO
The unemployment rate will be lower at the end of 2017 than at the end of 2016.	YES
If Hillary Clinton is elected President in 2016, the unemployment rate will be lower at the end of 2017 than at the end of 2016.	YES
Hillary Clinton sucks!	INVALID

Submit Report

FIG. 4. Sample Report. Reputation owners report on the *actual outcome* of an event, after the event has occurred. Each 'Your Report' entry is initially set to `NO REPORT ENTERED`; if submitted that way, the user will not receive credit for reporting on that event. `INVALID` reports are permitted – in fact, encouraged! – for poorly-worded and/or indeterminate questions, since truth-by-consensus works well only for outcomes which are easily and objectively determinable. Since this user's Reputation is 42, his/her ballot has a *weight* of 42. Another user who has a Reputation of 105 would also only cast a single ballot, but his/her opinions would be given 2.5 times as much weight as the first user's.

There are a few unusual things about the Report transaction. The first is the structured `quorum` field:

```
"quorum": {
    "matured": true,
    "reported": 0,
    "required": 2,
    "met": false
}
```

There are two requirements for *quorum* to be met:

1. The events being reported on must have passed their maturity time. Typically, if people are reporting on an event, this will be the case – events' maturity times should be set at or after the (expected) time of occurrence. In this example, the Report transaction has been made after the maturity time, as expected.

2. The minimum number of required reports must be met. In this simple example, there are only two reports required to meet quorum, zero of which have been provided so far.[10]

Once quorum is met, the market closes, and no more Bitcoin transactions are accepted. Report broadcasting continues for a pre-specified period of time. Upon completion of this phase, each Report is decrypted, and the Reports are gathered into a *report matrix*. This matrix has users as rows and events as columns; the values of the matrix are the users' observations.

The other unusual feature of the Report transaction is its output Script, which includes several new commands:

```
OP_DUP
OP_HASH160
<Jane's hash-160>
OP_EQUALVERIFY
OP_DATACHECK
OP_CONSENSUS
OP_PCACHECK
OP_EQUALVERIFY
```

The first, `OP_DATACHECK`, calculates the hash-160 of the Report's fields (excluding the report ID), then verifies that this hash matches the actual Report ID. This is to ensure that the transaction's contents were not tampered with during the delay from the time the Report was broadcast until the market closes.

The second new command, `OP_CONSENSUS`, initiates the consensus algorithm, which is described in detail in the following section. `OP_CONSENSUS` requires the report matrix as an input. The report matrix does not yet exist when the reports are first broadcast; instead, it is pieced together when the Reports are decrypted. Once the report matrix is assembled, it is written to the associated input's `scriptSig`.

`OP_PCACHECK` verifies that the outcome of the consensus algorithm is correct.[11] The final opcode,

[10] Not counting the current report, which has just been broadcast to the network, and is not yet included in a block.

[11] The consensus algorithm is a modified form of a common statistical technique called PCA (Principal Component Analysis) [18],

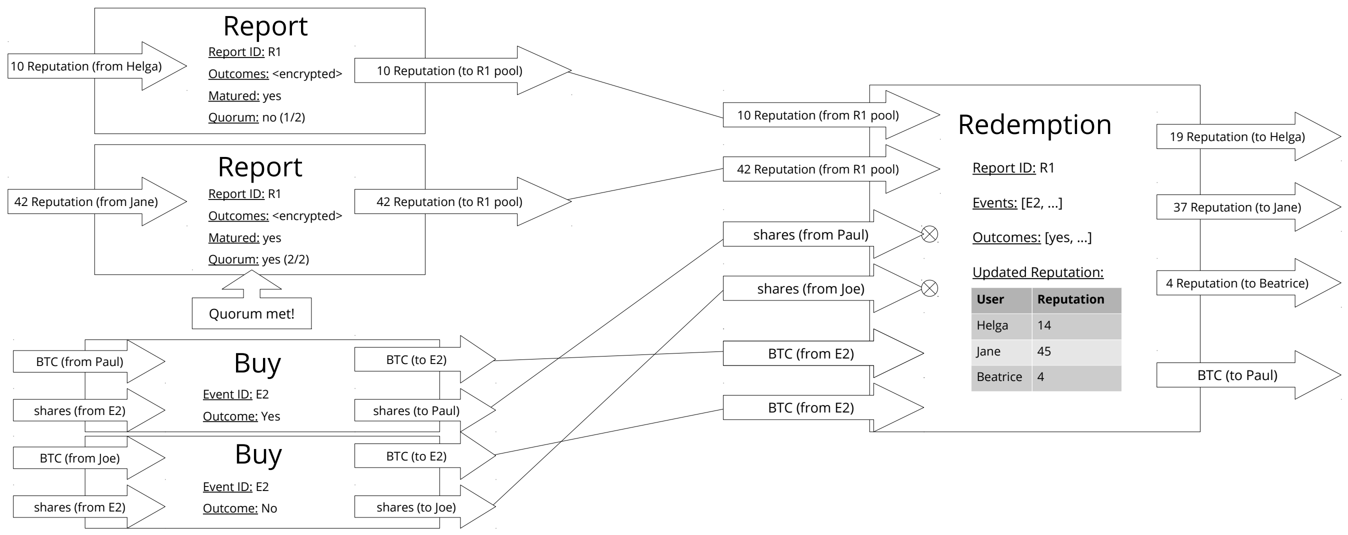

FIG. 5. Report and Redemption transactions. In this simple example, there are only 2 users reporting, Helga and Jane. (In reality, their names would not be known.) Helga broadcasts her report first, after the event has reached its maturity date. Since she is the first user to broadcast her report, only 1/2 users have reported, and quorum has not been reached, so the markets associated with the events listed on the ballot remain open. Later, Jane fills out her ballot and broadcasts it. At this point 2/2 users have reported, and quorum is reached. This automatically triggers the Redemption transaction, which makes use of special commands in the Report transactions' output Scripts to reach and verify consensus. Notice there are actually 3 users total; the third user, Beatrice, neglected to submit a Report, and therefore lost Reputation.

`OP_EQUALVERIFY`, compares the product with the original (centered) report matrix and ensures that they match.

2. *Redemption Transaction*

The Report output's matching input Script is as follows:

```
<Jane's public key>
<report matrix>
<centered report matrix>
```

The *centered report matrix* is the report matrix with the per-column weighted mean subtracted from each column.

This Script is found in the *Redemption* transaction (Fig. 5), a special transaction which is initiated when quorum is reached:

```
{
    "type": "Redemption",
    "vin": [
        {
            "n": 0,
            "value": 10,
            "units": "reputation",
            "scriptSig": "<Helga's public key>
                          <report matrix>
                          <centered report matrix>"
        },
        {
            "n": 1,
            "value": 42,
            "units": "reputation",
            "scriptSig": "<Jane's public key>
                          <report matrix>
                          <centered report matrix>"
        },
        <all outstanding shares>,
        <all outstanding wagers>
    ],
    "vout": [
        <all reputation>,
        <all wagers>
    ]
}
```

The Redemption transaction is quite large: it has two inputs for every outstanding wager – one for Bitcoin, one for shares.[12] Reputation balances are updated using a modified version of the *Sztorc consensus algorithm* [3].[13] This algorithm's purpose is to reward users whose reports are consistent with the consensus, and punish those whose reports are not.

Per-share values are fixed according to the consensus-determined outcomes. The fixed share-price values are then used to determine the payout sent to all users holding shares in these events. Payout values are assembled in a matrix containing the events in columns, and the

hence the name `OP_PCACHECK`.

[12] Various hard-coded size limits on transaction and block size are lifted specifically for the Redemption transaction.
[13] Details of the consensus algorithm are shown in Appendix A.

share owners' addresses in rows. When this payout matrix is complete, the market broadcasts it to the Augur network. Once it is incorporated into a block and added to the blockchain, the payouts appear in the recipients' accounts.

ACKNOWLEDGMENTS

The authors thank Vitalik Buterin, Paul Sztorc, Zack Hess, Alan Lu, Joe Costello, Jeremy Gardner, Casey Detrio, Joe Dolinak and Kinnard Hockenhull for their insights and feedback. Financial support for this project was provided by Joe Costello.

[1] C. Manski. Interpreting the predictions of prediction markets. *NBER Working Paper No. 10359*, 2004.

[2] J. Wolfers and E. Zitzewitz. Interpreting prediction market prices as probabilities. *NBER Working Paper No. 10359*, 2005.

[3] P. Sztorc. Truthcoin: trustless, decentralized, censorship-proof, incentive-compatible, scalable cryptocurrency prediction marketplace. *https://github.com/psztorc/Truthcoin*, 2014.

[4] M. Felson and R.V. Clarke. Opportunity makes the thief: practical theory for crime prevention. *Police Research Series, Paper 98*, 1998.

[5] U.S. Commodity Futures Trading Commission. CFTC charges Ireland-based "prediction market" proprietors Intrade and TEN with violating the CFTC's off-exchange options trading ban and filing false forms with the CFTC. Nov. 26, 2012.

[6] M. Philips. What's behind the mysterious intrade shutdown? *Bloomberg Businessweek*, Mar. 11, 2013.

[7] P.F. Yeh. Using prediction markets to enhance us intelligence capabilities: a "standard & poors 500 index" for intelligence. 50, 2006.

[8] S. Nakamoto. Bitcoin: a peer-to-peer electronic cash system. *https://bitcoin.org/bitcoin.pdf*, 2008.

[9] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timon, and P. Wuille. Enabling blockchain innovations with pegged sidechains. *http://www.blockstream.com/sidechains.pdf*, 2014.

[10] C. Slamka, B. Skiera, and M. Spann. Prediction market performance and market liquidity: a comparison of automated market makers. *IEEE Transactions on Engineering Management*, 60:169–185, 2013.

[11] D.M. Pennock, S. Lawrence, C.L. Giles, and F.A. Nielsen. The real power of artificial markets. *Science*, 291:987–988, 2001.

[12] R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Tech. Rep., George Mason University, Economics*, pages 1–12, 2002.

[13] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5:107–119, 2003.

[14] J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, 26(1):26–37, 1980.

[15] J. Skilling. Data analysis: the maximum entropy method. *Nature*, 309:748–749, 1984.

[16] S. Presse, J. Lee, and K.A. Dill. Extracting conformational memory from single-molecule kinetic data. *J. Phys. Chem. B*, 117:495–502, 2013.

[17] S. Presse, J. Peterson, J. Lee, P. Elms, J.L. MacCallum, S. Marqusee, C. Bustamante, and K. Dill. Single molecule conformational memory extraction: P5ab RNA hairpin. *J. Phys. Chem. B*, 118:6597–6603, 2014.

[18] J. Shlens. A tutorial on principal component analysis. *http://arxiv.org/abs/1404.1100*, 2009.

[19] G.R. Price. Extension of covariance selection mathematics. *Annals of Human Genetics*, 35:485–490, 1972.

[20] V. Buterin. Ethereum white paper: a next generation smart contract & decentralized application platform. *https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf*, 2013.

[21] G. Wood. Ethereum: a secure decentralised generalised transaction ledger, proof of concept VI. *http://gavwood.com/paper.pdf*, 2014.

[22] E.T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[23] E.T. Jaynes. Information theory and statistical mechanics II. *Physical Review*, 108(2):171–190, 1957.

[24] Principles of maximum entropy and maximum caliber in statistical physics. *Reviews of Modern Physics*, 85:1115–1141.

[25] J. Shore and R. Johnson. Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, 27:472, 1981.

Appendix A: Consensus Algorithm

The algorithm outlined here is a modified version of the *Sztorc consensus algorithm*, originally presented in [3]. In the original algorithm, the first eigenvector of the covariance matrix (the principal component) was solely used, regardless of the amount of variance it explained. Our modification is that a fixed *fraction of explained variance* is specified. This should make the rewards/punishments for reporting with/against the consensus more consistent across different Branches.

Suppose there are E total events being reported on by N users. We first take the ballot $\mathbf{B}$ of votes and construct its *weighted covariance matrix*, where the weights are Reputations. The centered ballot (the ballot with each event's weighted mean value subtracted out) $\mathbf{X}$ is:

$$\mathbf{X} = \mathbf{B} - \mathbf{1}\mathbf{r}^T\mathbf{B}, \tag{A1}$$

where T denotes the transpose operation and the difference is taken per-column. The ordinary (unweighted) covariance matrix is given by multiplying the centered data matrix and its transpose; similarly, the weighted covariance matrix is [19]:

$$\mathbf{\Sigma} = \frac{\mathbf{X}^T\mathbf{R}\mathbf{X}}{1 - \sum_i^N r_i^2}, \tag{A2}$$

where $\mathbf{R}$ has the elements of $\mathbf{r}$ on its diagonal and zeros elsewhere, $\mathbf{R}_{ij} = \delta_{ij} r_i$.

Each row (and column) of $\mathbf{\Sigma}$ corresponds to the variability *across* users, within a single event. Since there are E events, $\mathbf{\Sigma}$ is an $E \times E$ matrix.

Next, we diagonalize $\mathbf{\Sigma}$,

$$\mathbf{\Sigma} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^T, \tag{A3}$$

revealing $\mathbf{\Lambda}$, an $E \times E$ matrix with $\mathbf{\Sigma}$'s eigenvalues on its diagonal, and zeros elsewhere:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_E \end{bmatrix}. \tag{A4}$$

The eigenvalues are in descending order, $\lambda_i > \lambda_{i+1}$.

The eigenvectors of $\mathbf{\Sigma}$ are the columns of the similarity matrix, $\mathbf{S}$. The column of $\mathbf{S}$ associated with the largest eigenvalue (λ_1) is called the *principal component*, $\mathbf{s_1}$. This component $\mathbf{s_1}$ is a unit vector oriented in the direction of as much of the report matrix's variability as can be captured in a single dimension. This direction of maximum variability might be thought of as a hypothetical user who maximally contributed as much as possible to the variability in this particular set of reports.

The projection $\mathbf{p_1}$ of the centered data matrix ($\mathbf{X}$) onto the principal component $\mathbf{s_1}$ tells us how much each user's reports contributed to this direction of maximum variability. The same is true for $\mathbf{p_2}$: now the projection is onto the direction of next-largest-variability.

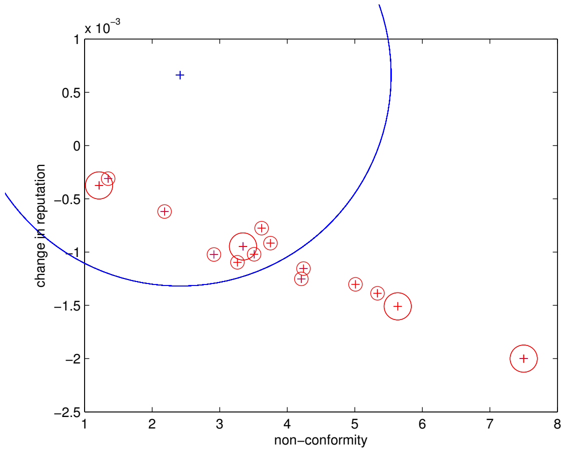

FIG. 6. Non-conformity scores plotted against users' change in Reputation after a single ballot resolution, in a small example of 50 users reporting on 25 events. The sizes of the circles indicate the number of users submitting identical reports. Blue indicates honest users; red indicates dishonest users.

Our strategy to achieve consensus is as follows. The cumulative fraction of variance (α_k) explained by the k^{th} component is given by [18]:

$$\alpha_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{j=1}^E \lambda_j}. \tag{A5}$$

Setting a fixed *variance threshold* (α) allows us to extract the number of components (n) needed to explain at least $\alpha \times 100\%$ of the report matrix's total variance. The *coordination vector* $\mathbf{c}$ is calculated by summing together the (centered) data projections onto the first n components, weighted by their eigenvalues:

$$\mathbf{c} = \mathbf{X}\sum_{j=1}^n \lambda_j \mathbf{s}_j. \tag{A6}$$

Element c_i of the coordination vector is an estimate of the *non-conformity* of user i. A high score indicates that this user contributed a great deal to the system's overall variability – that is, the user often disagreed with the consensus – and therefore was likely not honest. The user is therefore punished (Fig. 6).

The remainder of our consensus calculation, and Reputation redistribution calculation, is identical to the specification of [3].

Appendix B: Smart Contracts

A decentralized prediction market can be constructed in a relatively straightforward way using smart contracts, such as Ethereum [20, 21]. Due to the recent release of