Technologies that replace a "central planner"

By TOWNSEND, ROBERT M. AND ZHANG, NICOLAS X.*

Contracting possibilities are endogenous to the technologies on which the negotiation and the trading environments are built. In this paper we review how different technologies allow different implementations of the "central planner" from the mechanism design literature, and gradually abstract this centralized role away into different negotiation and trading environments. We illustrate how the "central planner" can be replaced by different institutional and technological arrangements that enable participating agents to, in sequential order: first carry out already agreed upon mechanisms without needing a "central planner" such as an auctioneer to see private bids; second to also conduct the negotiation for the contracting among themselves, without needing a "central planner" such as an auditor or a broker dealer who would need to access and escrow private balances; third to contract where previously impossible, and at scale where previously too costly. We emphasize that the "central planner" construct should no longer remain an abstract and mythical invocation in the literature, and that it can be (it already is, in contingent fashion) concretely deployed to implement optimized solutions to bilateral and multi-agent mechanism design problems[1]. We hope to provide inspiration and methods for economic, business and policy designers to be able to leverage the right technological tools to design the right solution to the specific problems they will be facing.

* Townsend: MIT Economics, 50 Memorial Dr, Cambridge, MA, rtownsen@mit.edu. Zhang: MIT CSAIL, 32 Vassar St, Cambridge, MA, nxyzhang@mit.edu.

[1]This paper is a high level summary of a more extended paper, *Innovative financial designs utilizing homomorphic encryption and multiparty computation*, Townsend and Zhang 2020. There we go over different implementations of more classical mechanism design problems with the full mathematical constructs, which we only describe at an intuitive level here

I. Letting agents carry out already agreed upon mechanisms without needing an auctioneer - a first layer of encryption to reduce what information trusted third parties need to see

In many mechanism design problems, agents want to conceal their private information from other agents involved, but they must provide these private information to some trusted third parties who will centrally execute the mechanism agents contracted into. The trusted third parties empowered with these private information can sometimes be tempted to abuse of their position[2], and are always exposed to higher risk of leakage or hacking[3]. Recent advances in privacy-preserving computations (described in the next paragraph) can however help tackle the treatment of the private information that is needed to execute a mechanism, so that the trusted third party only processes encrypted data, and can't decrypt them (and doesn't have the need to decrypt them), effectively reducing bad incentives and risks.

The key property of homomorphic encryption, HE, is that a function f can operate on a true underlying space or equivalently operate on the space of encrypted (i.e. encoded) values. That is, in order to utilize a function f in some application, one does not need to input the "plaintext" information which reveals the original and true values of a given agent, but rather, one can encrypt the agent's data, so that data can be kept private, apply the same function f on encrypted input data and get a result in the encrypted space. That is f(Enc(m)) = Enc(f(m)). Equivalently: Decipher[f(Enc(m))] = f(m). The distributivity allows operations outside the parenthesis to be "distributed" to the elements within, whereas a non distributive

[2]In 2017, Guardian sued the Rubicon project for the hidden fees it allegedly charged when running the auctions[**?**, **?**]

[3]unintentional data breaches such as those including Equifax, one of the biggest data breaches in US history [**?**]

scheme doesn't allow such treatment of the parenthesis. This allows us to perform a series of composed transitive functions g all on top of the encrypted message, such as: g[f(Enc(m))] = g[Enc(f(m))] = Enc[g(f(m))]. Equivalently decipher(g[f(Enc(m))]) = g[f(m)]. Thus, the agent or contract performing all the functions never gets to see the actual content of the message, nor the true outcome of f in the normal space

The key property of multiparty computation, MPC, is that to determine the value of a function f with inputs from multiple agents, one can run f on encrypted private inputs (with each private input being encrypted differently by each agent) and still get the correct output value of f after decryption. Namely, for J agents Decipher$[f(c_1,c_2,...,c_J)] = f(m_1,m_2,...,m_J)$ where $(c_1,c_2,...,c_J) = (Enc_1(m_1),Enc_2(m_2),...,Enc_J(m_J))$ are the encrypted values of the inputs from the J agents, each encrypted using a different key. Distributivity of MPC fails, and we can't compose transitive functions g on top of f - ie Decipher$[g(f(c_1,c_2,...,c_J))] \neq g(f(m_1,m_2,...,m_J))$. But combining MPC with HE as in Asharov, Jain, and Wichs 2012, we have MPC plus distributivity - ie the not equal sign above becomes an equality.

Overall, let's note that with HE and MPC, encrypted inputs can be kept private and still contribute to operations performed on top of them. We will call that a "private-but-contributing-state-of-information". What private information to reveal (or not) even as a computation is run on it, and who can decide on it, become a new choice element of mechanism design. For illustration, here below is a list of the main steps followed by each agent in a privacy-preserving computation such as that of de Castro et al 2020. Note that these steps (and the classical mechanism design problems example provided in the extended paper) are just specific implementations, but these tools can be tailored to other problems. We do not force applications onto a given technology, but go the other way around, starting with the economics, and then finding suitable technologies to implement them.

1) Each agent individually generates its own key pair, where each key pair contains a public encryption key and a private decryption key
2) All agents submit their public keys to each other (**classic MPC**) or to a central server or an agent chosen to perform the homomorphic operations (**HE+MPC version**).
3) This trusted party/server (**HE+MPC version**) or all agents through interactions (**classical MPC version**) combines all agents' public keys into a single joint/shared public key (**the core technique on which all forms of MPC are built**)
4) This new joint/shared public key is distributed to all agent
5) Each agent encrypts its private data using this new joint/shared public key, generating a ciphertext (an encrypted block of data) that cannot be decrypted by anyone but themselves
6) Each agent sends the ciphertext of its private data to the trusted party (**HE+MPC version**) or to each other (**classical MPC version**). This ciphertext completely hides the agent's data.
7) The trusted party (**HE+MPC version**) or all agents together through interactions (**classical MPC version**) run computations on all the encrypted data, producing an encrypted result of the computation (**that is the homomorphic encryption part**)
8) The encrypted result are sent back to each agent.
9) Each agent uses the private key they generated in Step 1 to partially decrypt the answer (which is still scrambled at that point)
10) Each agent sends this partially decrypted answer back to the trusted party (**HE+MPC version**) or to each other (**classical MPC version**). The trusted party (**HE+MPC version**) or all agents together through interactions (**classical MPC version**) recombine(s) the results of all the partial decryptions it/they received from all agents to produce the decrypted result that is then shared back with the agents (**in HE+MPC the server just needs agreement from k between 1 and J the total number of agents to decrypt. The value of k is chosen before any of these steps, and can follow mechanism design rules. In classical MPC all agents need to combine all partial decryptions to produce the decrypted result - which all J agents would see, which is not the case in HE+MPC**).

Figure 1: main steps followed in an HE-MPC computation

II. Letting agents also contract directly among themselves - by adding systematic audit, binding commitments and automatic enforcement

Let us assume that we now have a privacy-preserving version of an auction with private values (for full mathematical steps please see our extended paper). An astute reader can already raise from Figure 1 questions that could impact the mechanism, such as (A) what prevents an agent from bidding untruthfully (for instance by entering a bid higher than his endowment), especially since his inputs are encrypted? And (B) what would prevent him from reneging on his bid? To answer these we turn to distributed ledgers' ability to integrate on a platform both the payment functions and the contracting functions, so that (A) one can cap the amount one can bid to the endowment he has deposited onto this ledger - even if both endowment and bid are encrypted, and (B) the bid can be a transfer of the amount bidded from an agent to an escrow account managed only by the code implementing the auction (the "smart contract" - which is in effect a program replacing the auctioneer in this case) so that agents can't reneg on their bids.

In terms of cybersecurity, a distributed ledger also presents higher safety. For instance, even putting the design considerations mentioned just above aside, imagine a privacy-preserving auction using HE+MPC implemented through a program (doing the escrow, the audit and the releasing of goods and funds) running on a single server hosted at MIT. At any point in the auction process, anyone (including the operator of the server himself, as he can be bribed) could attack and maybe shut down this MIT server and interrupt the auction (even if that would still an improvement compared to many of today's auctions, as the operator of the server is a lesser version of an auctioneer that cannot see private inputs from participants). Running this code on servers hosted by professional computing third parties such as IBM, Amazon Web Services, Google Cloud and company would make it harder to attack or shut down the process, effectively relegating the "auctioneer" to an "infrastructure and code operator". Distributed ledger technologies are taking this one step further, as multiple parties (including the agents themselves) could become part of this "infrastructure and code operator", mitigating the bribing surface (most distributed ledgers are byzantine fault tolerant so more than 1/3 or half of the node operators need to be bribed, vs just one firm if delegated to a professional computing third party. Note that nodes in a distributed ledger can be run by different professional computing third parties!).

III. The dynamics of these new technologies in the music industry: a telling story

In this section we would like to argue that many of the changes we described - the gradual relegation of the role of "central planner" into more and more sophisticated, and less centralized, institutional and technological arrangements - are hard to avoid once the technologies we described mature. For this we would like to use the history of the music industry as an example (as music is a medium particularly prone to digitalization, and as it requires very few intermediaries. The same trends can also be observed in publishing, education, news, TV production... though in these industries more intermediaries are involved). In fact, the major records labels that were in charge of both scouting, recording/editing, and also distributing music until the early 2000s (acting as *physical* intermediaries prior to digitalization) had to partner with Internet streaming platforms once music was digitalized and the Internet permeated retail consumers. Internet streaming platforms such as Spotify effectively displaced records labels in the distribution of music, and also in scouting. Open access to these streaming platforms is allowing artists to upload their records without going through one of the major record labels, and the new data that these streaming platforms gathered also enabled new forms of scouting. Major records labels which used to be all-in-one intermediaries scouting, recording and distributing music now have to share some of these functionalities with the new digital platforms.

Now much criticism has been raised against these streaming platforms and the fees they extract from artists. The economies of scale that produced a few big (ferociously) competing

platforms further exacerbates this redistribution/wealth allocation problem. However, one of the brightest hopes for NFTs (or "pointers" in computer science terms) is that they can bundle a digital asset (such as a digitalized recording) to an address on a distributed ledger (which "points" to the digitalized recording). Encryption can restrict the capacity of the digitalized recording to be played to only the owner of the address on the distributed ledger (solving the problem of piracy - another form of the "double spending" problem that digital payment systems had to mitigate!), while code programmed into the NFT smart contract can fix ex-ante the royalties that an artist wishes to get from any current *and future* sales of this recording (eg from the sales of this address on the distributed ledger), along with as many parameters as the artist would wish (such as restricting the sales to only X number of people, to restrict resales only up to Y times, to restrict acquirer to be only of Z types... As long as one can code these into the contract!). These would take away a lot of the power online streaming platforms currently have, and relegate them to just "infrastructure and code operator", while artists themselves can become the "planner" of their music distribution.

IV. Implementing a "central planner" where previously too costly

One interpretation of the history of the music industry above is that before the advent of the Internet, only artists successful enough could catch the attention of records labels, which would then allocate to service this artist with technical staff, lawyers and distribution campaign resources. Online streaming platforms made distribution virtually free (as the early examples of pirate streaming platform shows), so a new category of artists could then benefits from the services provided by online platforms - the new intermediaries. However, because the platform still seeks to satisfy their needs first, the services offered to those artists not successful enough to be considered priorities are standardized at best, predatory at worst. Switching to NFTs to manage their IPs would be one way for these artists to take control of their career and tailor the distribution modalities according to their own preferences.

A parallel could be made in other industries. We would like to especially draw attention to financial services in low and middle income areas, as surveys such as the Townsend Thai Project highlighted: customers that don't bring enough value for traditional banks are left aside, so some Internet neobanks specialize in servicing them (Chime for instance for unbanked in the US, Robinhood for first time traders). Their services could be improved upon (even though day traders are the ones making the most losses on average, Robinhood still encourages them to trade as frequently as possible to satisfy its payment for order flow business model, for instance through nudging features and gamification). An open and programmable public infrastructure such as the one described in the IMF's "exchange and contracting platform" would on the other hand enable the deployment (sometimes by agents themselves) of automated contracts, at a cheaper cost (as the "exchange and contracting platform" is conceived as a public good). Programmability combined with automatic deployment also enable more granular and tailored products, similar to how Spotify enabled customers to create their own playlists, in contrast with the previous "one album for one artist" format. In the financial realm, such more granular and tailored products can be forecasted in securitization (as many loans are tokenized, and investors can write their own contracts to select automatically the loans fitting their preferences, and pool them together in a custom made security), and more broadly for any type of risk sharing contracts. The flexible contracts described in Townsend, 1989 and observed on the ground in Thai villages, where complex systems of credit history, flexible multi-temporal arrangements provide optimized risk sharing, can now be implemented as history of all sorts of interactions can also be recorded on a shared common ledger, with smart contracts to automate these processes without breaching individual privacy. This will further improve "impersonal" financial markets full of "foreign" entities not knowing each other and with no preexisting history nor future incentives, by porting in it more of the optimized but logistically expensive ("local") village markets. Just as how artists could get a more direct relationship with their audience by taking control of their distribution modalities!

V. Implementing a ”central planner” where previously too complicated - the ”exchange and contracting platform” to flexibly anchor the international monetary system

Flexible risk-sharing contracts as the ones described above can also be applied between countries and currencies - and even between cryptocurrencies, with or without fiat ones[4]. In fact, one motivation for the IMF's ”exchange and contracting platform proposal”[5] was to introduce a ”central planner” where previously impossible - in the design and updating of the international monetary system. Indeed, incentives and stakes are such that the design of an international monetary system (for instance at Bretton Woods) or the negotiations to update it (for instance at the Plaza hotel) are necessarily the results of lengthy negotiations, as private information have to be gradually distilled and digested among participants, for consensus to be reached. Multilateral privacy-preserving auctions running on the IMF's ”exchange and contracting platform” lets central banks regularly express in a privacy-preserving manner their reserves and FX preferences. Tying these two with automatic interventions acting where these preferences overlap, we are hoping to enable a systematic way in which central banks can collectively adjust the anchoring of different currencies, just as they would have done if there were a benevolent central planner in charge of harmonization between them. Central banks could for instance input the daily volatility bands they want their national currencies to stay in with regards to other currencies, and input the proportion of reserves in each of these currencies they are ready to commit to achieve this. These are like bids in a privacy-preserving auction, which would then after taking into account all central banks' preferences determine the ”grid” of exchange rates, along with the pooled central bank reserves to defend them (a smart contract could be in charge of using this pool to automatically lean against the wind when an exchange rate is exceeding its assigned volatility band).

This would be a more flexible version of the European Exchange Rate Mechanism (the ERM, or the ”snake in the tunnel”) that would allow participants to change at all time their goals and commitments, which could potentially prevent the issues that lead to the collapse of the ERM, and which would offer an interesting alternative to a currency union.

VI. Conclusion

We have illustrated here the extent to which different historic forms of economic contracts and organizations can be explained by information-incentive problems, and the technology available to them. These information technology related constraints are key in realizing how a seemingly imperfect form of organization was actually the best implementable possible, given the technologies of the time. Related, new technologies that allow relaxation of some of these constraints on the flow of information can potentially lead to better implementable mechanism design solutions. We have tried to convey that intuition here.

VII. Bibliography

1) Leo de Castro, Andrew Lo, Taylor Reynolds, Fransisca Susan, Vinod Vaikuntanathan, Daniel Weitzner, Nicolas Zhang. SCRAM: A Platform for Securely Measuring Cyber Risk. Harvard Data Science Review, 3:483-500, 09 2020

2) Robert M. Townsend , Tommaso Mancini Griffoli, Tobias Adrian, Federico Grinberg, Nicolas Zhang, A Multi-Currency Exchange and Contracting Platform, IMF Working Papers 2022/217, International Monetary Fund.

3) Robert M. Townsend. Information constrained insurance. Journal of Monetary Economics, 21:411–450, 03 1988.

4) Ari Juels. NFTs Can and Will Be So Much More, CoinDesk Opinion, 09 2022

[4]MIT LEAD also describes a decentralized implementation of this scheme for cryptocurrencies, where smart contracts privacy-preservingly aggregate beliefs and commitments, and use these (which no one knows as it's all encrypted) to arbitrage other crypto exchanges until the aggregated beliefs have been realized. This would anchor cryptocurrency prices and prevent sudden bubble-bursts of newly minted tokens - as to influence that token's price relative to other cryptocurrencies one would need to convince other holders to defend its price by committing other cryptos as reserves.

[5]Views are from the authors not that of the Fund