SoK: Privacy-Enhancing Technologies in Finance

Carsten Baum
Technical University of Denmark, Denmark

James Hsin-yu Chiang
Technical University of Denmark, Denmark

Bernardo David
IT University of Copenhagen, Denmark

Tore Kasper Frederiksen
Zama

Abstract

Recent years have seen the emergence of practical advanced cryptographic tools that not only protect data privacy and authenticity, but also allow for jointly processing data from different institutions without sacrificing privacy. The ability to do so has enabled implementations a number of traditional and decentralized financial applications that would have required sacrificing privacy or trusting a third party. The main catalyst of this revolution was the advent of decentralized cryptocurrencies that use public ledgers to register financial transactions, which must be verifiable by any third party, while keeping sensitive data private. Zero Knowledge (ZK) proofs rose to prominence as a solution to this challenge, allowing for the owner of sensitive data (*e.g.* the identities of users involved in an operation) to convince a third party verifier that a certain operation has been correctly executed without revealing said data. It quickly became clear that performing arbitrary computation on private data from multiple sources by means of secure Multiparty Computation (MPC) and related techniques allows for more powerful financial applications, also in traditional finance.

In this SoK, we categorize the main traditional and decentralized financial applications that can benefit from state-of-the-art Privacy-Enhancing Technologies (PETs) and identify design patterns commonly used when applying PETs in the context of these applications. In particular, we consider the following classes of applications: 1. Identity Management, KYC & AML; and 2. Markets & Settlement; 3. Legal; and 4. Digital Asset Custody. We examine how ZK proofs, MPC and related PETs have been used to tackle the main security challenges in each of these applications. Moreover, we provide an assessment of the technological readiness of each PET in the context of different financial applications according to the availability of: theoretical feasibility results, preliminary benchmarks (in scientific papers) or benchmarks achieving real-world performance (in commercially deployed solutions). Finally, we propose future applications of PETs as Fintech solutions to currently unsolved issues. While we systematize financial applications of PETs at large, we focus mainly on those applications that require privacy preserving computation on data from multiple parties.

2012 ACM Subject Classification Cryptography

Keywords and phrases DeFi, Anti-money laundering, MPC, FHE, identity management, PETs

Digital Object Identifier 10.4230/LIPIcs...

Funding *Carsten Baum*: Part of the work was carried out while the author was visiting Copenhagen University and supported by Partisia. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of Partisia.
Bernardo David: The project was supported by the Concordium Foundation, by the Independent Research Fund Denmark (IRFD) grants number 9040-00399B (TrA^{2}C), 9131-00075B (PUMA) and 0165-00079B, and by Copenhagen Fintech.
Tore Kasper Frederiksen: The work was carried out while at the Alexandra Institute, supported by Copenhagen Fintech as part of as part of the "National Position of Strength programme for Finans & Fintech" funded by the Danish Ministry of Higher Education and Science.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Modern Cryptography and Traditional Finance. Due to their sensitive nature, financial applications require strong security guarantees. Clearly, it is necessary to ensure authenticity and integrity of any financial operation, *i.e.* guaranteeing that the operation has been ordered by an entity authorized to do so and that this order has not been tampered with. Moreover, it is also necessary to achieve *privacy*, *i.e.* preventing attackers from obtaining sensitive information related to financial operations (*e.g. the identities of entities involved in a transaction and/or the value of that transaction*). In the digital realm, authenticity and privacy guarantees can be achieved against powerful adversaries who control communication networks (*e.g.* the Internet) by means of digital signatures and encryption, respectively.

A Decentralized Conundrum. The meteoric rise of decentralized financial applications based on cryptocurrencies and smart contracts hosted on blockchain platforms brought to light a whole new set of challenges. While traditional financial applications are hosted and executed by financial institutions in a centralized manner, the decentralized nature of blockchain-based applications requires all operations to be verifiable by third parties by means of publicly available records. If only simple cryptographic primitives are employed, this means that sensitive data that was once internally handled by financial institutions must now be exposed on the blockchain in order to perform a financial application. For example, Bitcoin requires revealing the sender and the receiver of a financial token that is transferred, so that a transfer transaction is considered valid if and only if the rightful owner of the token signs it.

Privacy-Enhancing Technologies (PETs) to the Rescue. While sacrificing privacy to achieve decentralization may be acceptable in some situations, most financial operations involving companies and private citizens cannot be conducted in this manner due to a number of reasons (*e.g.* protecting business interests and complying to regulations). In order to solve this issue, the cryptocurrency community turned to Privacy-Enhancing Technologies (PETs) that allow for achieving the same authenticity and privacy guarantees as in traditional centralized financial applications while providing the public verifiability guarantees needed in decentralized blockchain platforms. In particular, many of the first proposals towards this goal involved using a technology called Zero Knowledge (ZK) proof systems [75] : a method that allows the owner of sensitive data to prove a statement about this data without having to reveal it. For example, in the token transfer transaction example, a ZK proof allows a user to prove that an "encrypted" transfer transaction has been signed by the rightful owner of the token, without revealing neither the owner's nor the receiver's identity, *e.g.* as in [17].

From Decentralized to Traditional Finance. The vast usefulness of advanced PETs in blockchain applications also sparked an interest in deploying similar solutions for traditional financial applications. The aforementioned ZK proof technology has also been used in innovative solutions to the Know Your Client (KYC) and Anti Money Laundering (AML) problems commonly encountered in the banking industry, *e.g.*, as in [90, 132, 129, 124]. As in the case of privacy preserving cryptocurrency transactions, in many scenarios an entity wants to prove that they comply with KYC/AML regulations without revealing their identity nor their sensitive data. For example, a client can prove to a third party service provider that their identity has been verified by their bank and that they are authorized to use a certain service and perform operations up to a certain financial volume, while keeping their identity, and other attributes (*e.g.* the list of operations they are allowed to perform) private.

PETs for the Masses - or - From ZK to MPC. The ZK proof technology lends itself extremely well to applications that require a single entity to publicly prove a statement about its private data, *e.g.* the KYC/AML or cryptocurrency examples above. However, it is limited by the fact that the entity who generates a ZK proof must necessarily know all the private information about which the statement is proven. This is a serious limitation in two cases: 1. applications that must process sensitive data provided by multiple entities; 2. applications where certain data (*e.g.* cryptographic secret keys) are far too valuable to be stored on a single device, which leaks the data if its security is compromised. Fortunately, these limitations can be addressed by means of secure Multiparty Computation (MPC) [39, 74], which allows a set of entities to jointly execute an arbitrary program that computes on an "encrypted" version of their private data and only reveals the output of this computation.

For example, specific-purpose MPC protocols have long been used for sealed-bid auctions [80, 27] among entities who do not trust each other, nor a third party auctioneer. In this case, the parties provide as input "encrypted" versions of their bids and jointly compute a program that determines the winner of the auction, without revealing the value of the bids or any other information. In the context of blockchain-based cryptocurrencies, MPC protocols [88] have also been successfully deployed [47] for protecting secret signature keys used to authorizing/authenticating transactions. In this case, many entities locally stores a "share" of the signing key that does not reveal any information about the key itself unless all shares are united. When a transaction must be signed, all these entities use MPC to jointly execute a program that takes as input all the signing key shares, reconstructs the true key and computes the signature on the given transaction, while only revealing the resulting signature and nothing else. Since knowledge of the key is split among many entities, an attacker now has to compromise many, potentially all, entities instead of a single server.

1.1 Systematizing Privacy-enhancing Technologies in Finance

The goal of this SoK is to systematize financial applications that can benefit from PETs, as well as systematizing relevant PETs according to their respective applications and technological readiness. As summarized in Figure 1, we consider 5 main classes of financial applications, which are each addressed in the specific section indicated next to the application class name. At a high level, these applications can be potentially facilitated by the PETs indicated in the **PET** column of Figure 1, which are introduced in detail in Section 2. In particular, we focus on financial applications that require handling private data from multiple entities, as ZK proof technology for financial applications has been extensively addressed in previous works (*e.g.* [26, 4, 5, 103]). We summarize the main privacy enhancing technologies we consider in this SoK and their Technology Readiness Level (TRL) in Figure 2, recalling that the scale goes from 1 to 9 where level 1 means the basic principle has been observed and level 9 means proven successful in real-world applications. While we aim at providing a general overview of PETs for financial applications covering broad ranges of both PETs and applications, we do not intend to provide an exhaustive review of the PET literature. For each class of applications, we strive to survey the works that introduced the most relevant insights and groundbreaking results, since it would be infeasible to cover every single optimization of each PET that would be relevant for each application.

The financial applications we cover and the respective relevant PETs are summarized as follows:

Identity, KYC & AML (Section 3): Identity management is a classical problem that has the added challenges of Know Your Client (KYC) and Anti Money Laundering (AML) regulations in the financial sector. PETs can be used in these applications to provide robust identity management with privacy preserving methods for enforcing KYC & AML

Figure 1 PET stack for financial applications. TSS = Threshold Secret Sharing, DP = Differential Privacy, (F)HE=(Fully) Homomorphic Encryption, PSI = Private Set Intersection, MPC = Multiparty Computation, ZK = Zero Knowledge proofs. See Section 2 for a discussion of each concept.

	PET (§2)
(§3) Identity, KYC & AML	MPC, ZK
(§A) Legal	MPC, ZK
(§B) Digital Asset Custody	TSS, MPC
(§4) Markets & Settlement	(F)HE, MPC, ZK
(§5) Future applications	PSI, DP, MPC

Figure 2 Technology Readiness Level (TRL) between 1-9 of different PETs. See Section 2 for a discussion of each concept.

PET (§2)	**TRL**	*Demonstrated by*
ZK	9	ZCash [82], Filecoin [87]
TSS	9	Zengo [86]
DP	9	Apple [58]
PSI	8	Apple [18]
MPC	7	JP Morgen [50], Meta [38]
FHE	6	Zama [43]

regulations in both decentralized and traditional scenarios.

Legal Procedures (Section A): Many legal procedures require evidence to be presented in court. However, in many cases the evidence or even the relevant law/regulation must be kept private. PETs allow for such legal procedures to be conducted without sacrificing neither privacy nor auditability (*i.e.* the ability of any entity to verify that a legal procedure has been properly executed). Furthermore, due to the novelty of PETs it is not always clear how they fit within existing legal frameworks and how they might help and provide utility while fulfilling privacy regulations such as GDPR.

Digital Asset Custody (Section B): Digital assets such as cryptocurrencies are usually transferred by means of a digital signature, which can only be generated given a secret key. Since storing this key in a single device poses a risk of key leakage, PETs can be employed to distribute the signing power (and thus the power to move the asset) among many entities, in such a way that the system is only compromised if all entities are compromised.

Markets & Settlements (Section 4): Both the traditional and decentralized financial markets use complex trading instruments that may be abused by entities who retain privileged information about trades. PETs provide a robust solution to this issue via distributed "Dark Pools" or privacy preserving DeFi mechanisms (*e.g.* Automated Market Makers) that process trades without revealing any sensitive information to the entities evolved.

Future Applications (Section 5): Besides financial applications that have already been addressed in previous work, we propose that several PETs can be potentially used to address other interesting challenges in finance. In particular, recent advances in PETs enable the execution of advanced machine learning (ML) algorithms on private data, allowing for detecting patterns (*e.g.* for fraud) without revealing neither the ML models nor the data.

2 Available Privacy-enhancing Technologies

Before we describe applications of *Privacy-Enhancing Technologies* (PETs) to finance, we will give a short overview over existing PETs and how mature they are.

Zero-Knowledge proofs. Zero-Knowledge proofs [75] are cryptographic algorithms which allows a prover to convince a distrusting verifier that a certain statement is true. While the statement (usually specified in the form of a program) is known to the verifier, the proof (e.g. a certain input that makes the program output 0) is never leaked to the verifier. There exists a large variety of different ZK proof algorithms, and choosing the optimal proof depends largely on the application. Recently, efforts have been underway to standardize ZK proofs[1] to make them more accessible to practitioners.

Private Set Intersection. Private Set Intersection (PSI) [66] allows two (or more) distrusting parties with respective input sets S_1 and S_2 to securely learn their intersection, i.e. $S_1 \cap S_2$, without revealing the non-intersecting elements to the other party. For example, if party 1 has $S_1 = \{a, b, d\}$ and party 2 has $S_2 = \{b, c, e\}$ then both parties will learn that they have b in common in their sets. At the same time, party 2 will not learn that party 1 also had a, d in its input set and vice-versa for c, e. Highly efficient PSI protocols have been developed recently and some, such as the one developed by Chen *et al.* [42] found applications in industry.

Threshold Secret Sharing. Threshold Secret Sharing (TSS) allows a dealer to distribute [112] a secret x among n different parties, who each receive a *share* of the secret. Given a threshold $t < n$, TSS guarantees that if t or less parties pool their shares together, then they cannot reconstruct any information about x. If instead more than t parties cooperate (*i.e.* pool their shares), then x can be reconstructed. Multiple versions of secret sharing exist, for example with security against share-holders who don't act honestly during the reconstruction of the secret [45, 106]. Moreover, secret-sharing can be generalized so that not a threshold decides about the possibility of reconstruction, but instead any pattern can be used by the sender of the shares.

Multiparty Computation. Cryptographic protocols for Multiparty Computation (MPC) [16, 40, 73] allow 2 or more mutually distrusting parties who each have an input x_i to evaluate an arbitrary function $y = f(x_1, \ldots, x_n)$ on their inputs. MPC guarantees that only the function output y and no other information about the inputs is revealed. One can see PSI as a special case of MPC where the computed function is the intersection of input sets. MPC can be made robust against parties who maliciously deviate from the protocol description, and security usually holds if less than a threshold t of the participants in the computation collaborate to undermine the security. Therefore, MPC can be seen as constructing a *distributed trusted entity.* Recent progress in MPC research has made practical use of MPC possible[2].

Fully-Homomorphic Encryption. Fully-Homomorphic Encryption (FHE) is a special type of encryption scheme first proposed in [108] and later realized in [71]. In FHE, everyone with a so-called public key can encrypt information, while only the holder of the private key can decrypt it later. In addition, given encrypted of data as well as the public key, anyone can *perform computations* on the encrypted data and evaluate algorithms on secret inputs. For example. Given encryption $[x], [y], [z]$ of the values x, y, z, FHE allows to compute an encryption $[x \cdot y + z]$ of $x \cdot y + z$ or *any other efficiently computable algorithm* on these inputs. The clue is that the decryptor who obtains $[x \cdot y + z]$ will only learn $x \cdot y + z$ but not the *inputs to the computation.* Although concrete FHE schemes are relatively new, the technology is already somewhat mature[3] and powerful testing implementations[4] are available.

Differential Privacy. Differential Privacy [61] (DP) is a technique to compute *add noise to outcomes of algorithms* such that leakage about the inputs of the computation is minimized.

[1] See `https://zkproof.org/`.
[2] https://www.mpcalliance.org/
[3] https://fhe.org/
[4] https://www.openfhe.org/

The level of the noise is calibrated such that mathematical guarantees about the privacy of the inputs can be given.

A note on Trusted Execution Environments. Trusted Execution Environments (TEE) such as Intel's SGX are special modes of modern processors. A processor in its trusted execution setting guarantees that programs and their data are shielded from every other program running on the computer - even the operating system or any user having full access. A secure TEE allows to build many of the aforementioned PETs such as ZK proofs, PSI, MPC etc. "cheaply" and without additional cryptographic tools. In practice, SGX and similar technologies from other vendors[5] are regularly broken and do not offer the protection that they claim. We therefore do not consider it as a PET in this document.

3 Identity, KYC, AML

A general issue facing the financial world is the validation of customer identities and attributes. Laws and regulations require financial institutions, both classical and decentralized, to employ Know Your Customer (KYC) rules, for example to prevent money laundering and to be able aid in criminal cases - or even to lock accounts in case of sanctions. Being able to correctly determine a legal owner of an account can in itself help in preventing money laundering by precluding the use of fake accounts which could otherwise aid in *smurfing*, see Sec. 3.1. In the EU this is for example in place through the Anti-Money Laundering Directives and in the US through the Money Laundering Control Act of 1986. In the classical banking setting such validations are carried out through customers going to their physical bank and bringing required documents to prove their identity, residence or perhaps even criminal history, of which the bank would keep a copy. However, with the advent of online-only banks such as Lunar, Revolut and N26, along with crypto-currency exchanges like Binance and Coinbase, such validations become tricky, as there are no physical locations to validate identities.

Today KYC is instead carried out online, and in many cases through machine learning algorithms, where customers upload copies of their data which gets validated. When it comes to security this unfortunately as several disadvantages: i) it is easy to create a picture of a document or manufacture it, or even modify some data of a real document [118]; and ii) the leakage of legitimate documents online allows an adversary to steal identities. Simply considering how often a copy of ones passport is needed (e.g. basically any hotel or accommodation in any country), it is not hard to see that copies of legitimate documents will be easy to find on the dark market. Even though requirements can be made to include selfies or short videos to validate authenticity, this has turned into a race against Photoshop and deep fakes, which have shown tremendous advancement in the recent years [119]. While such attacks are also possible in physical space (i.e. creating fake documents and having them validated by a human), they are significantly more cumbersome due to human involvement and more expensive to mount, and therefore do not *scale* like digital-only attacks. Thus it is clear that the digital attack vector on KYC is the weakest link in account validation.

3.1 Identity management

One possible way of combating attacks when validating digital copies of physical identity documents is simply to move the documents into the digital realm. Digital signatures and revocation systems can ensure that digital documents are legitimate. This is done by

[5] See e.g. the exhaustive list on `https://sgx.fail/`.

combining them with an identification scheme where a user needs to prove they know a password/key used in the construction of their digital identity. This can prevent theft by simply copying the digital document. Often a simple password or key is not deemed secure enough in financial applications by law [102], and a second factor is required.Thus the use of authentication apps is common in electronic ID (eID) solutions, like the Danish MitID. Such eID solutions validate user-identities by the a trusted issuer during setup, allowing other applications to piggy-back on existing validation. This naturally comes with a risk of compromise or identity sharing through the eID provider, although it may arguably be harder than with their physical counterparts.

Single Sign-On. eIDs are typically validated by a centralized and trusted server that is able to perform relevant logging, and hence poses a risk to user privacy. Furthermore, such identity management is not exclusive to official or government identities, but can involve any kind of self-reported identity, which is the case for example for a Facebook or Google account. These platforms act as *federated identity management* services, allowing the sharing of the user's identity, along appropriate attributes of the user, to third-party websites. Thus facilitating a *single sign-on* (SSO) system. In this setting, the server validating the user's identity is known as the *identity provider* (IdP), which would be Facebook or Google in the above example. The third-party website is known as the *service provider*. This could for example be Netflix or Spotify. The idea of an SSO service goes beyond simply having an IdP facilitating a user authenticating towards a service provider. In fact an IdP may gather certified attributes about a user from multiple trusted issuers, and sign off on the user indeed being validated to have such attributed. This for example happens when Facebook validates that a given user has access to a specific email account, or phone number. While using an SSO makes things much simpler for a user, it also is a big privacy issue as an IdP now hold a large amount of the user's personal information, along with knowledge of whenever the user users this information and towards which service provider. Furthermore, it also means that large amount of trust has to be put in the IdP as they are would be able to impersonate any of their users towards any service provider. While such a thing is also possible for any attribute issuer (to a lesser extent) it becomes more of a problem for an IdP as they must be user-friendly enough that they can be used several times a day and since their only job is authentication. Hence becoming more exposed. Beyond this, simply using an SSO service can also lead to *traceability* and *linkability* of the user across the web. Traceability means that a user can be identified from the data resulting from using their eID. Whereas linkability means that it is possible for different service providers to find out if they have the same users. This can be an issue even if the user is authenticated using a pseudonym, since all it takes is one sharing of personal data, such as credit card information at a service provider, to de-anonymize the user. However, works like PASTA [2] and PESTO [14] use threshold cryptographic to enhance the security of IdPs and limit traceability and linkability without reducing the usability. I.e. password based authentication can still be used and they remain compliant with solutions like OAuth and OpenID Connect. While they only focus on password-based authentication, they can be generalized to support multi-factor authentication [65] and thus be used when multi-factor authentication is required for financial compliance, as e.g. in Europe according to PSD2 [102].

Decentralized Identifiers. With the advent of blockchain technologies a lot of work has sprung up, trying to remove centralization from the management of eIDs. This is generally known as a Decentralized Identifier (DID) [115]. The overall idea is that any kind of attribute provider issues a pseudonym to a user's blockchain account, reflecting a specific attribute. The user can then later use the pseudonym to prove certain certain attributes, or to simply get a reusable link to their pseudonymous identity at the same identity provider. However, it is clear to see that this basic construction is unfortunately not enough to ensure privacy, as again it possible to link the user across the internet (or blockchain) through their pseudonym. For this reason DID systems are starting to incorporate more advanced cryptographic constructions allowing users to anonymously prove that they hold a certain pseudonym towards a service provider (in order to facilitate authentication). Such a construction is known as a cryptographic *credential*.

Camenisch and Lysyanskaya [31] were the first to show a fully self-managed solution allowing users to prove their identity has been certified by a trusted provider, in an anonymous manner. Their credential construction affords validation of issuance from a trusted authority, while allowing the user to anonymously use it and preventing anyone who does not know the user's key[6] to impersonate it. However their construction did not allow the validation of arbitrary attributes. Something which is needed in many financial situations. Consider for example the case for loan or insurance issuance, where the customer's financial situation or health status has to be validated. Classically these must be provided as signed physical documents from the customer's attribute provider (such as credit bureaus), but the line of work on credentials, known as *attribute based credentials* shows how to achieve this in the digital sphere [32, 110] with cryptographic security and privacy guarantees. It was later shown how to compute arbitrary predicates on the certificated attributes [29]. Further development of such schemes into commercial products have been done by both IBM with their Idemix framework [33] and by Microsoft through U-Prove [100]. The underlying primitives have even been taken up by standardization frameworks such as W3C [114]. Still, despite such commercial traction, widespread adoption is still lacking. Moreover, in the context of DeFi systems, decentralized versions of anonymous credentials [70, 28, 6, 52] have been proposed.

One could imagine that the requirement for self-managed private keys could be the reason that such approaches lack adoption since the regular news bulletins of people having lost their cryptocurrency keys, show that self-administered key management is not for the general public. However, multiple solutions based on threshold cryptography can be used to securely store keys under a client's password [30, 83]. A more likely explanation might be the need of existing attribute providers to completely change their work-flow and systems, without any direct financial, legal or customer requirements.

Deploying Privacy Preserving Identity Management. Fortunately, recent research have shown how PETs can be used to get certified attributed from issuers without modifying existing infrastructure, when such attributes be retrieved from the provider through TLS-secured connections. The Town Crier system [131] shows how to construct certified attributes using secure hardware (like Intel SGX and using a TLS connection with a trusted provider). Concretely, they discussed how such certified attributes could be relayed to smart-contracts to allow more advanced decentralized user-attribute validation. Later, DECO [132] then showed how to remove the need for secure hardware and replace it with MPC while achieving the same goal. However, they extended their construction to also integrate with zero-knowledge proofs, allowing clients to construct certified proofs of arbitrary *predicates* on attributes from any provider, trusted through a TLS certificate which provides online access to the user's attributes. This could for example include a bank providing online banking access, where a user would then be able to construct a proof that they hold a bank account with e.g. at least \$20.000. If the user's government provides an online residency portal, then it could

[6] Allowing the user to fully control the use of their credential through a single key can be conceptually advantageous.

also be used for the users to prove that they legally reside in a given city in a given country without leaking their exact address.

The CanDID system [90] fully realizes a DID system with legacy support though either Town Crier or DECO. This is achieved through the usage of an MPC committee that validates legacy identity data and constructs a zero-knowledge friendly credential. Based on this credential, a user can prove arbitrary predicates on their attributes towards any provider.

Using attribute based credential allows the construction of fully private identity and attribute-based systems. However, in some situations full privacy might be undesirable, we would rather want to privately validate whether transactions are permissible based on attributes or identity, for example by ensuring that the identity of the credential holder is not on a deny-list. Kohlweiss *et al.* [85] showed that such a system can efficiently be constructed on top of credentials. The construction allows an auditor to specify any predicate on the attributes in a credential, where the identity of the credential holder gets leaked if the predicate is fulfilled. Such conditional privacy leakage could prove tremendously helpful in fighting money laundering as we discuss next.

Anti Money Laundering. Money laundering is the process of concealing the origins of money, such as financial gains from drug trafficking or other serious crimes, by changing its origin to a benevolent source. This is because criminals must acquire many services and goods in the regular economy: put simply, most luxury car dealers don't accept briefcases full of bills. Money laundering is a huge problem in the financial sector: the estimated amount of laundered money is at the level of 2-3% of the national GDP in the US alone, excluding tax evasion [107, Chap. 2].

Getting large amount of illegitimate cash into the financial system requires multiple steps and multiple accounts to avoid raising suspicion. Simply getting dirty money into the system is known as *placement*. A concrete and common approach for this is known as *smurfing*, where multiple legal people deposit small amounts of money for a criminal, with the promise of earning a small amount as a kick-back. After a period of time the smurfs move the money out of their accounts (minus their fee), by transferring to other accounts controlled by the criminal. If this process is done with small amounts, and the receiving party's account is not flagged, then smurfing is hard to identify[7].

Once the money is in the legal financial system, it needs to be mixed with legitimate transfers, to counter suspicions caused by the initial transfers. This involves creating reasonable and justifiable transfers among multiple accounts of multiple entities in a process known as *layering*. By setting up a layering scheme through multiple banks, in different legal jurisdictions, using different legal entities, it becomes almost impossible to trace the flow of money. This is because the involved banks are (reasonably!) not allowed to communicate private account and customer information about the sender and recipient of a money transfer. After the layering, the money is finally moved out of financial institutions and into legitimate investments such as real estate or legitimate businesses. This last step is known as *integration*.

What banks do to counter money laundering. As banks cannot share account and customer information with each other it is extremely hard for them to trace dirty money during layering. To address this, banks use multiple approaches usually subsumed as Anti-Money Laundering (AML) techniques. For example, banks internally use a *suspiciousness* score for customers. It is based on a *base* score, which is derived from the meta information about the account and its owner. The score may be derived from e.g. age of the account/holder, amount of

[7] This step is sometimes also realized through other means, such as deposits from cash-driven businesses such as laundromats or food trucks.

money in the account, expected income and nationality of the owner. Through transfers, the base score is then updated, e.g. based on the score of the account a transfer goes to or comes from if both sender and receiver account are held by the same bank. If they instead are held by different banks, then metadata such as the amount of money going in/out and the frequency of the transfers is used in updates.

Finally, banks do have one common tool in measuring the suspiciousness of transfers, and that is a common, yet secret, grey list. This grey list contains accounts that have been deemed significantly suspicious, but for whom no provable money laundering has been identified yet. A transfer to or from a grey-listed account significantly increases the suspiciousness score. At certain time intervals, the suspiciousness score of an account is checked against a certain threshold and if the score is too high, then it gets flagged for manual[8] inspection.

What can banks do? Due to GPDR and other privacy laws, it is not possible for banks to directly share meta-information about accounts or its owner without their consent. Furthermore, if a bank finds a flagged account it believes is engaging in illegal activities then when informing authorities, it must be able to *explain* to said authorities how they came to this conclusion. Hence the bank's judgements must be auditable by a third party. If the conclusion depends on data received from other financial institutions, the bank must be able to point to this data and the third party must trust it as well. While data from banks from within the same legal framework (the EU, USA, etc.) is usually considered as valid, data from international banks, in particular those from countries with a history of corruption, has less trustworthiness.

Implementing sufficient and efficient AML techniques is also difficult due to the quantities of information involved. AML technologies should ideally be scalable to include all transactions and accounts. At the same time, even a limited AML technology which only covers cross-country transfers or an arbitrary subset of accounts, could still make a substantial dent into the suspected large amount of money laundering currently going unnoticed.

Cryptography and AML. The conjunction of AML and MPC is new and the main bodies of work on the topic are by Zand *et al.* [129] and Egmond *et al.* [124]. Zand *et al.* show how computation on secret data can be used to notify an auditor of suspicious behavior. Egmond *et al.* show, in collaboration with multiple banks, how to use additively homomorphic encryption to obliviously update risk scores, and eventually (with consent from collaborating banks) decrypt the risk scores and flag accounts and customers appropriately.

However, related to this is the area of auditability of confidential transactions. As for example discussed by Tomescu *et al.* [121] where users are given a limited monthly "anonymity" budget. This budget is a certain amount of currency they are able to transfer anonymously per month. However, transfers surpassing this amount, is subject to deanonymization and clearance by a trusted auditor.

Finally, we note that a survey of real world concepts using PETs to combat financial crime has been conducted by the Future of Financial Intelligence Sharing consortium [92]. Unfortunately, many of their mentioned solutions require a trusted party to be involved.

As mentioned above, AML in centralized banking is challenging as the transaction graph is hidden due to e.g. privacy regulations. However in the decentralized finance space, such transaction graphs are usually visible. This is why most popular cryptocurrencies, such as Bitcoin, Ethereum or Cardano, are only pseudonymous and not anonymous[9]. Cryptocurrency

[8] In practice it turns out that about 95% of automatically flagged accounts are false-positives.

[9] We note that there exist privacy-focused blockchains like ZCash, Monero, or Dash that hide the transaction graph. Moreover, one can build private transactions on top of non-privacy focused blockchains

exchanges such as Coinbase and Binance allow to turn large amount of cryptocurrency into Fiat currencies. These exchanges are required by law to enforce know-your-customer (KYC) rules. Through the help of transaction graph analysis firms such as Chainalysis, it has become hard to launder money using pseudonymous cryptocurrencies.

Researchers have also proposed mechanisms to enforce AML even if transactions are kept private. This includes using an escrow system where anonymity and privacy can be broken in case suspicious activities occur, such as transfers to or from an account known to be used by criminals [109, 98, 10, 52]. Such escrow mechanisms does not necessarily imply the usage of a trusted third party, as the data for escrow activities can e.g. be shared using Threshold Secret Sharing. Another approach is to specify a small budget per client which they can use every month for anonymous payments. After the client has made more transactions than covered by this budget, any future transactions can be traced [127, 121]. Although seemingly a good compromise between privacy and security, this does still pose a risk to smurfing.

4 Markets & Transaction Settlement

In this section, we systematize PETs in market and settlement applications.

In financial markets, there is a need for auctions and markets with *fairness* guarantees, as rational actors are incentivized to collude and front-run honest parties, if the true valuation or trade-intent of the latter is revealed. Here, we first consider the *traditional finance* setting (§4.1), where the settlement of transactions is handled by traditional, asynchronous settlement processes. In the presence of a *public ledger* (§4.2), settlements occur *synchronously* and *immediately* after a transaction is completed. Such a mechanism also permits the "netting" of inter-bank payments §4.3) to minimize the liquidity requirements on participating banks; this must done with PET approaches, since the public ledger would otherwise expose all individual payment orders, a clear breach of consumer privacy.

Finally, we highlight approaches to achieve bidder privacy in *demand-response* electricity markets (Appendix C), which coordinate the remote scheduling of power consuming devices to match forecast production from sustainable production sources; the submission of granular device-level information to an auction in the clear can reveal the activity and presence of customers at home, violating their privacy.

4.1 Markets in Traditional Finance

The first setting reflects an idealized view of *traditional finance*, where accounts and balances are generally maintained by financial institutions and considered private. Here, the settlement of auctions, or exchange transactions, occur asynchronously; whilst the *counterparty risk* from defaulting on obligations implied by pending transactions is real, we consider it an orthogonal challenge addressed in the public ledger setting (§4.2, §4.3). Clearing prices and executed volumes are considered public information as this information is forwarded to institutions executing the settlement. This first setting intends to achieve resilience against dishonest venue operators and participants attempting to obtain a financial gain from unwarranted information flow. Communication between parties generally assumes direct, authenticated channels, implying the knowledge of identities and a pubic key infrastructure.

using e.g. Zether [24] that leverages encryption and ZK proofs. Finally, mixers such as Tornado [105] take transfers from many users and put them into a holding account, from which they can later be transferred to the intended recipient.

Setting	Applications		Privacy	Benchmarks	PET	Works
Markets in Traditional Finance (§4.1)	Distributed sealed-bid auctions	Single-sided	Bid privacy	○	MPC	[64], [80], [27], [97]
		Double-sided	Bid privacy	◐	MPC	[21], [20]
		Public verifiability	-	○	HE+ZK	[101]
	Distributed Dark Pools	Continuous matching	Order privacy	◐	MPC	[36]
		Periodic matching	Order privacy	●	MPC	[37], [48]
			Order privacy	◐	FHE	[9]
		with many assets	Order privacy	●	MPC+HE	[37]
		with many servers	Order privacy	●	MPC	[48]
		Public verifiability	-	○	HE+ZK	[120]
Markets on Public Ledgers (§4.2)	Decentralized sealed-bid auctions	Single-sided	Order privacy	○	MPC+ZK	[8], [54], [68]
	Privacy-preserving Decentralized Exchanges	Futures Periodic matching	Net position privacy	◐	MPC+ZK	[91]
		Periodic matching	Balance privacy Partial order privacy	◐	MPC+ZK	[77]
			Order privacy (Balance privacy)	◐	MPC+ZK	[13], ([12])
		Intent-based order matching	Balance privacy Order privacy	◐	ZK	[22], [128]
			Balance privacy Order privacy	●	WKA	[99]
Settlement on Public Ledgers (§4.3)	Liquidity preserving inter-bank netting	-	Payment privacy	◐	ZK	[35]
			Payment privacy & Robustness	◐	MPC+ZK	[55]
Demand-Response Markets (*Appendix C*)	Distributed auctions for demand flexibility	Double-sided (Single buyer)	Device power constraint privacy	◐	MPC	[1], [133] [67]

Figure 3 Auctions & Markets: no benchmarks (○), preliminary benchmarks (◐), benchmarks achieving real-world performance with traditional market parameters (●).

Distributed Sealed-bid Auctions. One-off, sealed-bid auctions are frequently performed in the sale of frequency-spectrum rights, government contracts, real-estate and other private items, such as art. In open-cry, single-sided auctions, bids are broadcast publicly until no additional bids are made. However, the leakage of bids or orders can be exploited by the adversary for financial gain. In Vickrey auctions, where the winner pays the price submitted by the second-highest bid, the auction operator collecting the submitted bids is incentivized to collude with other bidders to increase the second-highest bid price and maximize auction fees. The auction operator must also be *trusted* to not reveal anything about submitted bids, in order for all bidders to submit their *true valuation.* The advent of the public, commercial internet coincides with the first protocol proposals which permit the execution of one-time, sealed-bid auctions by distributing the role of the auction operator, thereby removing the need for a trusted auction venue.

Franklin et al. [64] propose a one-sided auction protocol, where the auction venue is distributed amongst multiple servers; bidders to submit their signed bids as *verifiable secret-shares* (VSS) to participating servers during the bidding phase. Subsequently, bids and signatures are jointly reconstructed by all servers, upon which all bid information becomes public. Verifiable secret-sharing ensures that bidders submit well-formed bids. As long as a single server is honest, the reconstruction of bids cannot occur before the end of the bidding phase. However, it is often important to protect the privacy of bids, even if they are not successful; the valuation may reveal a bidding strategy for another, related auction.

In the work of Harkavy *et al.* [80], MPC is deployed to maintain the privacy of all submitted bids; only the winning bid is made public. Naor *et al.* [97] propose a variant of MPC with garbled circuits, to reduce the rounds of communication, thereby improving performance. Cachin [27] builds a purpose-built, privacy-preserving protocol, which permits the comparison ($>$) of *prices* between two parties with the help of an untrusted third party. An auction determining the highest bidder is then constructed from this primitive.

The first work to demonstrate the feasibility of privacy-preserving (one-time) *double-sided, sealed-bid auctions* was proposed by Bogetoft *et al.* [21]. Later secure auctions were used in practice [20], to facilitate the auctioning of sugar beet delivery contracts in Denmark. Concretely, farmers producing sugar beets hold contracts which represent an obligation and right to deliver beets to the the (single) Danish sugar beet processor Danisco. The trading of such contracts permits the reallocation of contracts to the most efficient producers, but such an exchange run by Danisco would permit it to learn information about the economic circumstances of producers, potentially compromising sugar beet farmers during contract negotiations. The matching and determination of a price computation from 1229 buy and sell orders was achieved in approximately half an hour by an MPC committee of 3 servers; a throughput volume sufficient for a one-time auction, but unacceptable for traditional electronic security exchanges. Notice, however, that cryptographic techniques have improved drastically since this work. Such an auction would run much more efficiently today.

Publicly verifiable auction operators are proposed in the work of Parkes *et al.* [101], a weaker alternative to implementing the auction operator with MPC; instead, the dark pool venue is still operated by single entity, but provides cryptographic proofs that the auction algorithm is performed correctly by the venue operator. Whilst this prevents the auction venue from *manipulating* the correct evaluation of auction bids, it does not prevent the auction operator from leaking bid information to malicious participants.

Distributed Dark Pools. Dark pools have gained adoption in traditional finance as venues where submitted orders are not publicly accessible, thus minimizing the potential price impact caused by signalling *trade intent* to front-running market participants. As is the case with auction venues, the dark pool operator must be trusted to not *share* the order flow information with malicious participants, a trust assumption that is frequently violated in practice (Table 1 in [37]), motivating the need for distributing the role of the venue operator.

Whilst the matching of orders in a *secret* order book is a natural domain of MPC, we observe that current proposals illustrate specific configurations and architectures for distributed dark pools that may or may not match throughput observed in traditional, centralized dark pool markets. In particular, the choice of auction clearing algorithm remains a deciding feasibility factor. Whilst secret-sharing based MPC schemes permit secret computation with n participating servers, the runtime bottleneck generally lies in the amount of *communication* that is required between servers. This is because MPC has a specific model of defining computations, and certain auction clearing algorithms can be realized with less communication overhead in MPC than others.

In the case of auction clearing algorithms, which must compute a *clearing price* from current bids and sell orders, the *sorting* thereof by *price limit* induces many *comparisons* $(>, <, =)$ between secret-shared values, which in turn imply sub-protocols generating the majority of communication cost. Alternatively, order matching based on volume only (where prices are determined by third party price feeds) can greatly accelerate throughput, as the expensive clearing price evaluation is not required. Furthermore, whether orders are processed *continuously* or *periodically* also greatly affects the real-world applicability of the following distributed dark pool protocols.

Continuous Double Auctions with MPC: A recent line of work by Smart *et al.* [36, 37, 48, 49] implements and examines real-world, double-side auction algorithms. In the initial work [36], continuous double auctions (CDA) are implemented in a distributed fashion across servers running an MPC. Continuous double auctions maintain a limit order book (LOB) where buy and sell orders are ordered by ascending and descending price respectively; *each incoming order* is matched against one or more LOB orders if it crosses the "spread" between best buy (or sell) prices; its remaining volume is then inserted into the LOB. It is also the most *expensive* exchange algorithm since (1) each single order must be matched against m other fulfilled orders and (2) its remaining trade volume must be inserted into a (potentially large) order book of N size. Benchmarking such an algorithm requires specifying the *expected* state of the order book, given the sensitivity of CDA run-time on (1) the average number of matched orders m; and (2) the expected order book length N. In the dark CDA implementation of Cartlidge *et al.* [36], run with 3 servers and Shamir-sharing based MPC, a worst-case throughput of $34 - 43$ orders per second for LOB parameters $m = 3$ and $N \approx 30$ is achieved. This work demonstrates that distributed CDA with MPC cannot yet match the throughput volumes of traditional CDA venues[10]. In contrast, *periodic* order matching greatly improves the performance of distributed dark pools.

Periodic Double Auctions with MPC: Periodic auctions in the dark pool setting implemented with MPC promise throughput that match those of traditional dark pool markets, as shown in these works by Cartlidge *et al.* [36, 37]. In periodic auctions, limit orders are submitted during an open auction period after which a clearing price is computed during the clearing phase, which maximizes the volume of matched orders (unmatched orders are carried over to the next round). In contrast to CDA algorithms, where orders are processed individually against a potentially large order book, periodic auctions only need to compute a single clearing price for the entire batch in a given period. In fact, real-world order execution throughput has been achieved with a realistic number of asset pairs.

In [37], the London Stock Exchange Group's *Turquoise Plato Uncross*, a widely-used traditional dark pool supporting thousands of assets, is implemented with promising results. Based on volume observed on the real-world Turquoise Plato Uncross venue, it is assumed that order book clearing occurs at most every 5 seconds, where at most 2000 newly input orders must be processed across an asset universe of 4000 financial instruments. This throughput was successfully handled by smaller MPC committee sizes of 2 (dishonest majority) and 3 (honest-majority), but required multiple MPC instances, each handling orders trading a small subset of all assets (Figure 4). For example, ~ 280 MPC committee instances are each randomly assigned 16 assets in each round by a *gateway* engine. This gateway periodically reassigns asset subsets to new MPC instances in order to break potential linkages between orders across time periods. We note that auction algorithms are not entirely *oblivious.* Indeed, the direction of orders are leaked in [36, 37], whilst volumes and order limits remain private.

Complementary follow-up work by Da Gama *et al.* [48, 49] both focus on (single-asset) privacy-preserving *volume matching*, which enables further performance gains since the clearing prices are determined by an external reference price; [48] introduces an improved MPC volume matching algorithm which permits dummy orders and hides the trade direction. [49] scales volume matching up to MPC instances consisting of ~ 100 servers and shows

[10] In their work, the runtime of MPC pre-processing is neglected, which represents a non-trivial "hidden" computational cost that can be performed during "offline" hours or outsourced to dedicated pre-processing workers; pre-processing generally does not limit the maximum, sustainable throughput of MPC.

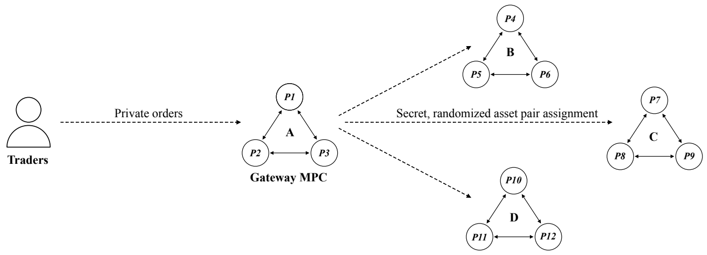

Figure 4 In [37], a gateway MPC (A) distributes inbound received orders across multiple MPC committees (B,C,D) to improve order clearing throughput. MPC servers never learn the asset pairs its committee is assigned in each round.

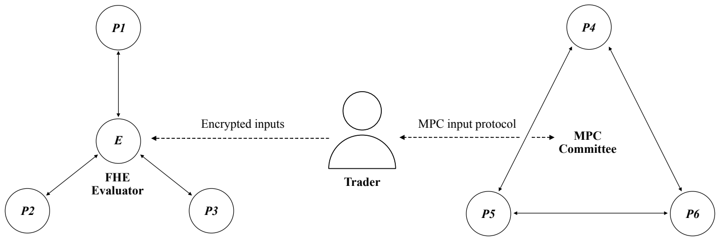

Figure 5 In [9], market auctions are implemented with fully homomorphic encryption (FHE); here, the encryption key is jointly generated by key servers (P1-P3), but the FHE evaluation is solely performed by the evaluator, who never learns the plaintext of the inputs or intermediary results. Decryption of the FHE output requires interaction with all key servers. In contrast, private computation with MPC in [36] requires interaction amongst MPC servers (P4-P6) for each (multiplicative) operation.

the economic costs associated with operating such a single server in a MPC instance of up to 100 servers to be below $\sim$ 0.10 USD and $\sim$ 0.025 USD for computation and network communication respectively in each auction round; the negligible cost demonstrates the feasibility of *market participants* contributing to the distributed operation of dark pools.

Periodic Double Auctions with FHE: JP Morgan has demonstrated initial results in work by Balch *et al.* [9] to realize dark pool venues where the venue operator is not distributed, but computes the periodic volume matching over data encrypted under a jointly controlled public key; here, the secret encryption key material used in the *threshold fully homomorphic encryption* is held in secret-shared form by all participants (Figure 5). While the computation can be done by one party on the ciphertexts (not knowing their plain values), the participants then later take part in a so-called distributed decryption protocol which reconstructs the outcomes to the venue operator. Whilst [9] benchmark periodic volume matching implemented with threshold FHE, the omission of the *partial decryption sub-protocol* complicates the evaluation of its performance. Still, FHE offers an alternative approach to secret sharing-based MPC which promises competitive performance; fewer communication rounds are required, since computation is performed locally by the dedicated venue operator over encrypted data, although local computation (over encrypted data) is more costly.

Publicly, verifiable dark pool operator: Similar to verifiable one-sided auctions [101], a weaker notion of order privacy for dark pools is proposed in the following work by Thorpe *et al.* [120], where the venue operator only reveals a homomorphically encrypted order book to traders; the operator itself, however, maintains the encryption key and can thus compute over the order book plaintext. Each update to the public, encrypted order book triggered by a submitted order is accompanied with a zero-knowledge range proof generated by the operator; any public party can locally re-compute the claimed update over encrypted values and verify zero-knowledge range proofs that guarantee that the plaintext values lies within certain ranges, thereby enabling verification of comparison statements between encrypted values. This system ensures the correctness of each order book update without revealing the order details themselves. [120] implements the CDA algorithm in a publicly verifiable manner; as in [101], this approach does not prevent any misuse of the order information held by the operator.

4.2 Markets on Public Ledgers

The advent of public ledger protocols [69, 84, 72] resulting from permissionless participation of servers across the public internet promises a truly "server-less" system of transaction settlements, no longer dependent on any single trusted intermediary. The state of the ledger is public and its integrity is publicly verifiable (by any online party) by local verification of all previously finalized transactions sequenced in form of a append-only list or *blockchain.* The realization of a global transaction history also implies a Turing-complete *state machine*; smart contracts represent user-deployed programs run on blockchain protocols that, in addition to custom ledgers [ERC20, ERC721], can realize decentralized auctions or decentralized exchanges (DEX), which forgo the need for trusted venue operators. In contrast to traditional finance, market applications in the public ledger setting offer instant settlement; any market application implemented with smart contracts instances permits the simultaneous evaluation and settlement between participants. Despite scalability challenges arising from the vast number of participants running the blockchain backbone protocol, the promise of instant settlement would allow the mitigation of counter-party risk, a real cost to transactions conducted in traditional finance today.

However, the public verifiability of a public ledger also introduces novel challenges for financial applications; account balances are public by default and leak information about submitted bids, trades or margin positions; the latter must be backed by valid balances. In decentralized finance (DeFi) [126], front-running is indeed rampant in decentralized exchanges (DEX) [122], since pending transactions leak trade intent to the adversary which can precisely order and inject transactions to execute optimal front-running strategies. Thus, proposals have been made to implement private balances on public ledgers with publicly verifiable, non-interactive zero-knowledge [111, 25]. However, a privacy-preserving ledger (even with standard smart contract support) is generally *not sufficient* for privacy-preserving financial applications such as exchanges [15].

Privacy-preserving ledgers generally complicate the realization of *smart contracts*, since these must verify and update account balances known only to its *owners* according to an agreed-upon transition logic. For decentralized exchanges implemented in the privacy-preserving ledger setting, this requires the presence of a secure multiparty computation instance, to which users can privately input their trade orders and private balances; the MPC then computes an updated DEX state and private balances, which are then updated on the ledger (Figure 6). Enforcing consistency between the *secret*, internal MPC state and *private* account balances on the ledger requires protocol design advances illustrated in the subsequent paragraphs. We emphasize that counter to popular belief, zero-knowledge is not sufficient to realize universally expressive, privacy-preserving smart contracts, as the witness (or secret state) for decentralized privacy-preserving applications are partially held by separate, distrusting parties; instead, function evaluation over private inputs from separate parties and secret-shared data is the natural domain of secure multiparty computation.

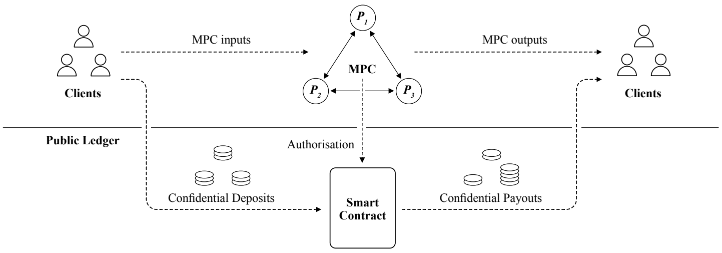

Figure 6 We sketch the architecture of privacy-preserving smart contract applications in MPC with instant settlement on a (confidential) ledger; clients provide input parameters to the MPC instance, and forward financial deposits to a smart contract in a confidential manner. The MPC privately returns computation output to clients, but also authorizes a new financial distribution which is paid out to the clients by the smart contract functionality.

We note there are privacy-preserving smart contract proposals which shield *private data* [117, 116] held by individual users or *private contract logic* [22], but such techniques are generally limited in their expressiveness. The work of Bowe *et al.*[22] only supports two parties, and is not widely used to realize privacy-preserving financial applications.

Sealed-bid Auctions (with Instant Settlement). The first work by Bag *et al.* [8] to realize sealed-bid auctions specifically in the setting of public ledgers focuses on using the blockchain as a *communication medium* instead of a settlement layer; as a permissionless protocol, any party can anonymously post an arbitrary message to the bulletin board, visible to all other parties. For protocols with low communication rounds, this is a practical solution; in particular, the simplicity of evaluating single-sided sealed-bid auctions permits task-specific secure multiparty protocols which only require public message broadcasts. The SEAL [8] protocol proposes the use of a *anonymous veto protocol* [79] requiring only two communication two rounds, that is then repeated once by auction bidders for *each bit* of their bid price, thereby removing the necessity an auctioneer role entirely. In its particular, the veto protocol of [79] receives the private input $b_i \in \{0,1\}$ for party $i \in [n]$, for each of the m bits representing the permissible price range; the parties learn the highest bid, bit by bit. Each execution of the veto protocol will thus publicly output 1 if one of the users submits a veto; thus, by repeating the veto protocol for each bit position, all participants receive the highest bid without revealing the prices of failed bids. [8] assumes participants to behave according to the protocol (semi-honesty).

This mechanism was later adopted and hardened by FAST [54] to be secure against malicious participants not adhering to the protocol. Furthermore, [54] introduces *guaranteed settlement* on the public ledger featuring privacy-preserving deposits. Participants are thus committed to execute the payment for their bids if these are successful during the auctions. Cheating participants are penalized by having their deposits slashed and reimbursed to other parties; non-interactive zero-knowledge proofs from all parties ensure that parties only submit a veto if it is consistent with their initial bid (in commitment form on the ledger); still, despite the privacy-preserving aspects of the anonymous veto protocol, privacy leakage occurs when the highest bidder learns when he overtakes the second highest bidder. The work of Ganesh *et al.* [68] adopts a similar construction which is proven to be game-theoretically secure; it is rational to pursue the honest protocol despite any strategy chosen by other players. The work of Chin *et al.* [44] does not employ zero-knowledge proofs to shield commmitted funds for a single-sided, sealed-bid auction; deposits are sent to committed, yet undeployed contracts and are thus indistinguishable from normal Ethereum transactions, a similar technique used in Breidenbach *et al.* [23] to commit inputs to smart contracts without revealing them to the front-running adversary. This approach only guarantees k-anonymity and relies on the presence of other, unrelated transactions.

Privacy-preserving Decentralized Exchanges. We note a number of recent proposals for privacy-preserving DEX applications in recent years; intent-based privacy-preserving DEX applications mirror the functionality of over-the-counter (OTC) venues (in traditional finance) and only require a public ledger, but do not scale well and are not widely deployed. Privacy-preserving and front-running secure DEX protocols generally involve private ledger deposits and perform the order matching in an MPC instance, as is the case in distributed Dark Pool proposals previously described in Section 4.1, but offer instant settlement following each DEX round (Figure 6).

Intent-based, privacy-preserving DEX. In the works of [22] and [99], a simpler model of a decentralized exchange is implemented; a bulletin board functionality provided by a public ledger permits a "maker" to broadcast their trade intent. An interested counter-party or "taker" then directly opens an authenticated communication channel with the maker to jointly perform a privacy-preserving atomic swap on the public ledger [22]. Intent-based DEX protocols resemble over-the-counter models in traditional finance. [99] introduces a "witness key agreement" (WKA) construction which preserves the privacy of the maker's offer; the WKA allows a taker to establish a shared secret key with a maker which has posted its order in commitment form to the ledger. The key agreement protocol succeeds if the committed, private order fulfills a relation determined by the taker. This key permits subsequent anonymous communication with the maker to finalize the transaction.

Privacy-preserving Futures DEX. An interesting example of decentralized exchanges is illustrated by Massacci *et al.* [91], which realizes a futures exchanges modelled closely after the Chicago Mercantile Exchange; here, the *future obligation* (or contract) to buy or sell a commodity is traded. The net position of a market participant is the sum of both current liquidity balances and future obligations; importantly, a party holding a future to "sell" a given commodity, must always hold sufficient liquidity to acquire the respective commodity, as it otherwise would default on its contractual obligation. Thus, a net position that falls below zero must be liquidated to protect the counter-party of any futures contract held by the liquidated party. Achieving this in a privacy-preserving manner without *revealing the net position* of a party is the goal of the work of [91].

If the net position of a participant is revealed, price manipulations could be conducted with the explicit intention of forcing the liquidation of otherwise valid positions. Thus, [91] proposes a similar scheme to [111], where the net position of each account is committed in a cryptographic accumulator. The validity of each update to the account is proven in zero-knowledge, whilst the trading venue is executed in a MPC instance, similarly to the Dark Pool proposals in Section 4.1. [91] requires parties participate in the protocol for each account update, even if this means the liquidation of their own account. We note that the subsequent privacy-preserving smart contract framework instantiated with MPC and a confidential ledger [12] achieves the privacy guarantees of [91] without permitting users to block application liveness.

Front-running Secure DEX. A general motivation for privacy in decentralized exchanges is the front-running of DEX applications in Decentralized Finance due to public transactions and accounts in the default ledger setting; Despite offering instant settlement of trades and transactions, pending user input authorizations generally broadcast a users trade intent

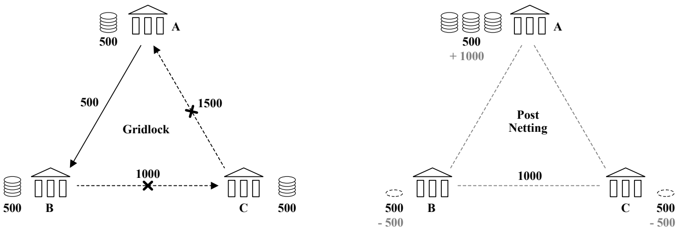

Figure 7 We adopt a netting example from [125]; processing of individual payment orders may fail due to a lack of liquidity (left), as balances must remain positive following execution of each individual payment. Netting relaxes this constraint; balances need only to be positive following execution of all payments orders (right).

before their finalization. To this end, P2DEX [13] proposes the first privacy-preserving decentralized exchange, which can operate and settle transactions across multiple ledger instances; clients submit orders to an MPC committee which computes the order matching and subsequently settles these on the respective public ledgers; since the trade inputs are private, front-running is mitigated. Follow-up work [12] generalizes this model to a setting with confidential accounts; here, all zero-knowledge proofs are moved *outside* the MPC computation, as computing such proofs *inside* the MPC remains generally unfeasible for real-world application. The work of Govindarajan *et al.* [77] realizes a privacy-preserving DEX in a similar manner; here, however, the actual order matching is computed in the clear of a smart contract over anonymized trade lists to accelerate the determination of a clearing price.

4.3 Inter-bank Netting on Public Ledgers

Inter-bank payment requests are currently submitted to the real-time gross settlement (RTGS) system managed by the central bank to update the accounts of sending and receiving financial institutions. In times of low liquidity, a bank may fail to honor *individual payment instructions*, as the liquidity requirement may exceed its balance and credit line granted by the central bank; a *gridlock* occurs, when a failed payment settlement prevents further payment instructions from being processed. Given the large payment volumes processed each day, liquidity saving mechanisms are implemented which settle payment instructions on a *netting basis* (Figure 7).

Recent work has proposed *distributing* the role of the RTGS operator with a public ledger protocol, whilst implementing efficient netting protocols with smart contracts [125, 96], therby increasing system resiliency as the operational liability burden on the central bank operator today is very high. Whilst the aforementioned works implement inter-bank netting of queued payments, the nature of public ledgers means that payment instructions are revealed to parties participating in the underlying blockchain backbone protocol. Instead, [35] proposes payment instructions to be posted to the ledger in *commitment form* accompanied with non-interactive zero-knowledge proofs attesting their well-formedness. Here, local netting solutions are computed *by each participating bank* and verified by a coordinating smart contract, which verifies correctness of all submitted, local netting solutions without revealing amounts and the identity of institutions. Since parties must compute partial netting solutions, the protocol of [35] is not *robust* against cheating participants, who can stall or abort the netting process by posting invalid partial netting proposals. In contrast, [55] computes the netting solution inside an MPC instance, thereby achieving *fault tolerance* against dishonest participants.

Despite initial implementation benchmarks provided by works above, it remains an open question in what configuration such systems can scale to real-world payment settlement volume and what netting frequency is required in practice.

5 Future applications

We will now outline which other PET use cases could be of interest in the financial sector in the foreseeable future. While many of the use cases previously described in this work may also not yet be production-ready, we want to highlight areas in this section which we think deserve more attention by researchers and practitioners. This necessarily is of speculative nature, so the reader may see this as food for thought.

Voting. Voting is a standard mechanism in deciding on future policies. While in many cases it is sufficient to make the whole voting process public, this is not always possible. For example, a voter may fear repercussions or embarrassment if his or her vote becomes public. Hence, to ensure *honest digital voting*, cryptographic voting algorithms have to be used. These ensure that election outcomes can be computed while individual votes cannot be attributed to participants. While such cryptographic voting can be realized using MPC or FHE, a dedicated line of work started by Chaum [41] presents highly efficient dedicated voting protocols. Cryptographic voting mechanisms find interesting applications in the DeFi space, e.g. for privacy-preserving Decentralized Anonymous Organizations (DAOs). In particular, a treasury system for DAOs based on electronic voting has been proposed in [130] and a board room voting scheme based on smart contracts (and this amenable to the DAO scenario) has been proposed in [93]. We believe that these techniques may also be useful for coordination among classical banking institutions and other financial operations (*e.g.* shareholder meetings).

Fraud detection. Both insurance and gambling are known as industries where companies in the sector exchange information on their customers in order to detect fraud or exploitative customers[11]. This information sharing may be problematic for privacy reasons, and it also leaks information about suspected but ultimately honest customers if done in plain. PETs such as PSI might be an interesting tool to construct a trusted intermediary. This intermediary can obtain information from participating companies and alerts them if e.g. more than 3 of them share the same customer. Here, PSI can ensure that only those customers are revealed that appear often enough.

Better and fairer pattern recognition. In section 3 we have outlined how AML does benefit from recognition of suspicious patterns. Such patterns, if one wants to keep up in the digital age, must be learned from a large dataset and must be updated frequently. Moreover, many companies in an industry have an interest in pooling their data with other institutions for the purpose of learning these patterns. At the same time, they may not want to share raw customer or transaction data. Another, related area is assessing the credit risk of potential customers. Here, the risk becomes more accurate the more participants can contribute information or models. At the same time, input providers have an interest to keep their data private (for data protection or to protect intellectual property).

Both applications fall into the area of privacy-preserving Machine Learning [89, 60, 95] or confidential benchmarking [51] which are subfields of MPC. While these areas have received much attention recently[12], optimized applications to finance seem to be lacking.

[11] For example the infamous “Griffin Book”.

[12] Privacy-preserving Machine Learning opens up interesting use cases, but it does not come without its

Another important aspect is that (automatically generated) models should not be biased against certain groups. While fair machine learning itself is a rapidly developing field, its application to finance [56] may deserve more attention.

Privacy preserving mitigation of systemic risk. Audits of financial institutions guarantee that their balances plus credit cover outstanding obligations. This reduces counter-party risk and means that the overall system can rely less on biasable methods such as ratings and reputation. At the same time, an audited company may not want to open its books fully to the public, or it might not be guaranteed that these books are correct. [91] have shown how audits can be realized using ZK proofs, although limited to the futures market. We believe that this concept may be generalized to the wider financial system to permit privacy-preserving audits.

References

1 Aysajan Abidin, Abdelrahaman Aly, Sara Cleemput, and Mustafa A Mustafa. An mpc-based privacy-preserving protocol for a local electricity trading market. In *Cryptology and Network Security: 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings 15*, pages 615–625. Springer, 2016.

2 Shashank Agrawal, Peihan Miao, Payman Mohassel, and Pratyay Mukherjee. PASTA: PASsword-based threshold authentication. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 2042–2059. ACM Press, October 2018. doi:10.1145/3243734.3243839.

3 Ignacio Alamillo, Cristina Timon, and Julian Valero. Oblivious identity management for private user-friendly services: D3.2 security and privacy-aware olympus framework impact assessment, 2020. URL: https://olympus-project.eu/wp-content/uploads/2020/02/Olympus_pu_d3_2_v1_0.pdf.

4 Ghada Almashaqbeh and Ravital Solomon. Sok: Privacy-preserving computing in the blockchain era. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 124–139, 2022. doi:10.1109/EuroSP53844.2022.00016.

5 Nasser Alsalami and Bingsheng Zhang. Sok: A systematic study of anonymity in cryptocurrencies. In *2019 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–9, 2019. doi:10.1109/DSC47296.2019.8937681.

6 Elli Androulaki, Jan Camenisch, Angelo De Caro, Maria Dubovitskaya, Kaoutar Elkhiyaoui, and Björn Tackmann. Privacy-preserving auditable token payments in a permissioned blockchain system. In *AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020*, pages 255–267. ACM, 2020. doi:10.1145/3419614.3423259.

7 Gilad Asharov and Claudio Orlandi. Calling out cheaters: Covert security with public verifiability. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 681–698. Springer, Heidelberg, December 2012. doi:10.1007/978-3-642-34961-4_41.

8 Samiran Bag, Feng Hao, Siamak F Shahandashti, and Indranil Ghosh Ray. SEAL: Sealed-bid auction without auctioneers. *IEEE Transactions on Information Forensics and Security*, 15:2042–2052, 2019.

9 Tucker Balch, Benjamin E Diamond, and Antigoni Polychroniadou. SecretMatch: inventory matching from fully homomorphic encryption. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–7, 2020.

10 Amira Barki and Aline Gouget. Achieving privacy and accountability in traceable digital currency. Cryptology ePrint Archive, Report 2020/1565, 2020. https://eprint.iacr.org/2020/1565.

own problems. See [57] for a good overview.

11 David A. Basin, Ralf Sasse, and Jorge Toro-Pozo. The EMV standard: Break, fix, verify. In *2021 IEEE Symposium on Security and Privacy*, pages 1766–1781. IEEE Computer Society Press, May 2021. doi:10.1109/SP40001.2021.00037.

12 Carsten Baum, James Hsin-yu Chiang, Bernardo David, and Tore Kasper Frederiksen. Eagle: Efficient Privacy Preserving Smart Contracts. *Cryptology ePrint Archive (To appear in Financial Cryptography and Data Security 2023)*, 2022. https://eprint.iacr.org/2022/1435.

13 Carsten Baum, Bernardo David, and Tore Kasper Frederiksen. P2DEX: privacy-preserving decentralized cryptocurrency exchange. In *International Conference on Applied Cryptography and Network Security*, pages 163–194. Springer, 2021.

14 Carsten Baum, Tore Kasper Frederiksen, Julia Hesse, Anja Lehmann, and Avishay Yanai. PESTO: proactively secure distributed single sign-on, or how to trust a hacked server. In *IEEE European Symposium on Security and Privacy, EuroS&P 2020, Genoa, Italy, September 7-11, 2020*, pages 587–606. IEEE, 2020. doi:10.1109/EuroSP48549.2020.00044.

15 Carsten Baum, James Hsin yu Chiang, Bernardo David, Tore Kasper Frederiksen, and Lorenzo Gentile. Sok: Mitigation of front-running in decentralized finance. Cryptology ePrint Archive, Paper 2021/1628, 2021. To appear on the Proceedings of the The 2nd Workshop on Decentralized Finance (DeFi) in Association with Financial Cryptography 2022. URL: https://eprint.iacr.org/2021/1628.

16 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988. doi:10.1145/62212.62213.

17 Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. doi:10.1109/SP.2014.36.

18 Abhishek Bhowmick, Dan Boneh, Steve Myers, Kunal Talwar, and Karl Tarbe. The apple PSI system, 2021. https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf. URL: https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf.

19 Dor Bitan, Ran Canetti, Shafi Goldwasser, and Rebecca Wexler. Using zero-knowledge to reconcile law enforcement secrecy and fair trial rights in criminal cases. In Daniel J. Weitzner, Joan Feigenbaum, and Christopher S. Yoo, editors, *Proceedings of the 2022 Symposium on Computer Science and Law, CSLAW 2022, Washington DC, USA, November 1-2, 2022*, pages 9–22. ACM, 2022. doi:10.1145/3511265.3550452.

20 Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, and Tomas Toft. Secure multiparty computation goes live. In Roger Dingledine and Philippe Golle, editors, *FC 2009*, volume 5628 of *LNCS*, pages 325–343. Springer, Heidelberg, February 2009.

21 Peter Bogetoft, Ivan Damgård, Thomas Jakobsen, Kurt Nielsen, Jakob Pagter, and Tomas Toft. A practical implementation of secure auctions based on multiparty integer computation. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006*, volume 4107 of *LNCS*, pages 142–147. Springer, Heidelberg, February / March 2006.

22 Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu. Zexe: Enabling decentralized private computation. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 947–964. IEEE, 2020.

23 Lorenz Breidenbach, Phil Daian, Florian Tramèr, and Ari Juels. Enter the hydra: Towards principled bug bounties and {Exploit-Resistant} smart contracts. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1335–1352, 2018.

24 Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. In Joseph Bonneau and Nadia Heninger, editors, *FC*

2020, volume 12059 of *LNCS*, pages 423–443. Springer, Heidelberg, February 2020. doi:10.1007/978-3-030-51280-4_23.

25 Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. In *International Conference on Financial Cryptography and Data Security*, pages 423–443. Springer, 2020.

26 Joseph Burleson, Michele Korver, and Dan Boneh. Privacy-protecting regulatory solutions using zero-knowledge proofs: Full paper, 2020. URL: https://a16zcrypto.com/privacy-protecting-regulatory-solutions-using-zero-knowledge-proofs-full-paper/.

27 Christian Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 120–127, 1999.

28 Jan Camenisch, Manu Drijvers, and Maria Dubovitskaya. Practical UC-secure delegatable credentials with attributes and their application to blockchain. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 683–699. ACM Press, October / November 2017. doi:10.1145/3133956.3134025.

29 Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 345–356. ACM Press, October 2008. doi:10.1145/1455770.1455814.

30 Jan Camenisch, Anja Lehmann, Anna Lysyanskaya, and Gregory Neven. Memento: How to reconstruct your secrets from a single password in a hostile environment. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 256–275. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_15.

31 Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001. doi:10.1007/3-540-44987-6_7.

32 Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_4.

33 Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Vijayalakshmi Atluri, editor, *ACM CCS 2002*, pages 21–30. ACM Press, November 2002. doi:10.1145/586110.586114.

34 Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. Uc non-interactive, proactive, threshold ecdsa with identifiable aborts. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1769–1787, 2020.

35 Shengjiao Cao, Yuan Yuan, Angelo De Caro, Karthik Nandakumar, Kaoutar Elkhiyaoui, and Yanyan Hu. Decentralized privacy-preserving netting protocol on blockchain for payment systems. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 137–155, Cham, 2020. Springer International Publishing.

36 John Cartlidge, Nigel P Smart, and Younes Talibi Alaoui. MPC joins the dark side. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 148–159, 2019.

37 John Cartlidge, Nigel P Smart, and Younes Talibi Alaoui. Multi-party computation mechanism for anonymous equity block trading: A secure implementation of turquoise plato uncross. *Intelligent Systems in Accounting, Finance and Management*, 28(4):239–267, 2021.

38 Benjamin Case, Richa Jain, Alex Koshelev, Andy Leiserson, Daniel Masny, Thurston Sandberg, Ben Savage, Erik Taubeneck, Martin Thomson, and Taiki Yamaguchi. Interoperable private attribution: A distributed attribution and aggregation protocol. Cryptology ePrint Archive, Paper 2023/437, 2023. https://eprint.iacr.org/2023/437. URL: https://eprint.iacr.org/2023/437.

39 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (abstract) (informal contribution). In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, page 462. Springer, Heidelberg, August 1988. doi:10.1007/3-540-48184-2_43.

40 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *20th ACM STOC*, pages 11–19. ACM Press, May 1988. doi:10.1145/62212.62214.

41 David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

42 Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1243–1255. ACM Press, October / November 2017. doi:10.1145/3133956.3134061.

43 Ilaria Chillotti, Marc Joye, and Pascal Paillier. Programmable bootstrapping enables efficient homomorphic inference of deep neural networks, 2020. https://whitepaper.zama.ai/whitepaper.pdf. URL: https://whitepaper.zama.ai/whitepaper.pdf.

44 Kota Chin, Keita Emura, Kazumasa Omote, and Shingo Sato. A Sealed-bid Auction with Fund Binding: Preventing Maximum Bidding Price Leakage. In *2022 IEEE International Conference on Blockchain (Blockchain)*, pages 398–405. IEEE, 2022.

45 Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th FOCS*, pages 383–395. IEEE Computer Society Press, October 1985. doi:10.1109/SFCS.1985.64.

46 Aloni Cohen and Sunoo Park. Compelled decryption and the fifth amendment: Exploring the technical boundaries. *Harvard Journal of Law and Technology*, 32(1):169–233, 2018.

47 Coinbase. Coinbase to acquire leading cryptographic security company, Unbound Security, Nov 2021. https://www.coinbase.com/blog/coinbase-to-acquire-leading-cryptographic-security-company-unbound-security.

48 Mariana Botelho da Gama, John Cartlidge, Antigoni Polychroniadou, Nigel P Smart, and Younes Talibi Alaoui. Kicking-the-bucket: Fast privacy-preserving trading using buckets. In *International Conference on Financial Cryptography and Data Security*, pages 20–37. Springer, 2022.

49 Mariana Botelho da Gama, John Cartlidge, Nigel P Smart, and Younes Talibi Alaoui. All for one and one for all: Fully decentralised privacy-preserving dark pool trading using multi-party computation. *Cryptology ePrint Archive*, 2022. https://eprint.iacr.org/2022/923.

50 Mariana Botelho da Gama, John Cartlidge, Nigel P Smart, and Younes Talibi Alaoui. Privacy-preserving dark pools. *Cryptology ePrint Archive*, 2022. https://eprint.iacr.org/2022/923.

51 Ivan Damgård, Kasper Damgård, Kurt Nielsen, Peter Sebastian Nordholt, and Tomas Toft. Confidential benchmarking based on multiparty computation. In Jens Grossklags and Bart Preneel, editors, *FC 2016*, volume 9603 of *LNCS*, pages 169–187. Springer, Heidelberg, February 2016.

52 Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. Balancing privacy and accountability in blockchain identity management. In Kenneth G. Paterson, editor, *CT-RSA 2021*, volume 12704 of *LNCS*, pages 552–576. Springer, Heidelberg, May 2021. doi:10.1007/978-3-030-75539-3_23.

53 Ivan Damgård, Thomas P Jakobsen, Jesper Buus Nielsen, Jakob Illeborg Pagter, and Michael Bæksvang Østergaard. Fast threshold ecdsa with honest majority. *Journal of Computer Security*, 30(1):167–196, 2022.

54 Bernardo David, Lorenzo Gentile, and Mohsen Pourpouneh. FAST: fair auctions via secret transactions. In *International Conference on Applied Cryptography and Network Security*, pages 727–747. Springer, 2022.

55 Angelo De Caro, Andrew Miller, and Amit Agarwal. Privacy-Preserving Decentralized Multi-Party Netting, September 29 2022. US Patent App. 17/216,644, https://patents.google.com/patent/US20220309492A1/en.

56 Leo de Castro, Jiahao Chen, and Antigoni Polychroniadou. Cryptocredit: securely training fair models. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

57 Emiliano De Cristofaro. A critical overview of privacy in machine learning. *IEEE Security & Privacy*, 19(4):19–27, 2021. doi:10.1109/MSEC.2021.3076443.

58 Apple Differential Privacy Team. Learning with privacy at scale. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf. URL: https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf.

59 Whitfield Diffie and Susan Landau. The export of cryptography in the 20th century and the 21st, 2005. URL: https://privacyink.org/pdf/export_control.pdf.

60 Wenliang Du, Mikhail J Atallah, et al. Privacy-preserving cooperative scientific computations. In *csfw*, volume 1, page 273, 2001.

61 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 265–284. Springer, Heidelberg, March 2006. doi:10.1007/11681878_14.

62 European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

63 Fireblocks. Fireblocks. Institutional Digital Asset Custody, Settlement & Issuance, Nov 2021. https://www.fireblocks.com/.

64 Matthew K Franklin and Michael K Reiter. The design and implementation of a secure auction service. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1996.

65 Tore Kasper Frederiksen. A holistic approach to enhanced security and privacy in digital health passports. In Delphine Reinhardt and Tilo Müller, editors, *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021*, pages 133:1–133:10. ACM, 2021. doi:10.1145/3465481.3469212.

66 Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 1–19. Springer, Heidelberg, May 2004. doi:10.1007/978-3-540-24676-3_1.

67 Mariana Gama, Fairouz Zobiri, and Svetla Nikova. Multi-party computation auction mechanisms for a p2p electricity market with geographical prioritization, 2022. https://www.esat.kuleuven.be/cosic/publications/article-3526.pdf.

68 Chaya Ganesh, Bhavana Kanukurthi, and Girisha Shankar. Secure Auctions in the Presence of Rational Adversaries. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1173–1186, 2022.

69 Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 281–310. Springer, 2015.

70 Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In *NDSS 2014*. The Internet Society, February 2014.

71 Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. doi:10.1145/1536414.1536440.

72 Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

73 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th FOCS*, pages 174–187. IEEE Computer Society Press, October 1986. doi:10.1109/SFCS.1986.47.

74 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987. doi:10.1145/28395.28420.

75 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. doi:10.1145/22145.22178.

76 Shafi Goldwasser and Sunoo Park. Public accountability vs. secret laws: Can they coexist? Cryptology ePrint Archive, Report 2018/664, 2018. https://eprint.iacr.org/2018/664.

77 Kavya Govindarajan, Dhinakaran Vinayagamurthy, Praveen Jayachandran, and Chester Rebeiro. Privacy-preserving decentralized exchange marketplaces. In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2022.

78 Lev Grossman. Inside apple CEO tim cook's fight with the FBI. *Time*, 2016. Accessed on 31/01/2022. URL: https://time.com/4262480/tim-cook-apple-fbi-2/.

79 Feng Hao and Piotr Zieliński. A 2-round anonymous veto protocol. In *International Workshop on Security Protocols*, pages 202–211. Springer, 2009.

80 Michael Harkavy, J Doug Tygar, and Hiroaki Kikuchi. Electronic auctions with private bids. In *USENIX Workshop on Electronic Commerce*, 1998.

81 Lukas Helminger and Christian Rechberger. Multi-party computation in the GDPR. *IACR Cryptol. ePrint Arch.*, page 491, 2022. URL: https://eprint.iacr.org/2022/491.

82 Daira Hopwood, Sean Bowe†, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification, 2021. https://zips.z.cash/protocol/protocol.pdf.

83 Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. TOPPSS: Cost-minimal password-protected secret sharing based on threshold OPRF. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17*, volume 10355 of *LNCS*, pages 39–58. Springer, Heidelberg, July 2017. doi:10.1007/978-3-319-61204-1_3.

84 Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*, pages 357–388. Springer, 2017.

85 Markulf Kohlweiss, Anna Lysyanskaya, and An Nguyen. Privacy-preserving blueprints. *IACR Cryptol. ePrint Arch.*, page 1536, 2022. URL: https://eprint.iacr.org/2022/1536.

86 Team KZen. Bitcoin wallet powered by two-party ECDSA extended abstract. https://github.com/ZenGo-X/gotham-city/blob/master/white-paper/white-paper.pdf. URL: https://github.com/ZenGo-X/gotham-city/blob/master/white-paper/white-paper.pdff.

87 Protocol Labs. Filecoin: A decentralized storage network, 2017. https://filecoin.io/filecoin.pdf.

88 Yehuda Lindell and Ariel Nof. Fast secure multiparty ecdsa with practical distributed key generation and applications to cryptocurrency custody. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1837–1854, 2018.

89 Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 36–54. Springer, Heidelberg, August 2000. doi:10.1007/3-540-44598-6_3.

90 Deepak Maram, Harjasleen Malvai, Fan Zhang, Nerla Jean-Louis, Alexander Frolov, Tyler Kell, Tyrone Lobban, Christine Moy, Ari Juels, and Andrew Miller. CanDID: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In *2021 IEEE Symposium on Security and Privacy*, pages 1348–1366. IEEE Computer Society Press, May 2021. doi:10.1109/SP40001.2021.00038.

91 Fabio Massacci, Chan Nam Ngo, Jing Nie, Daniele Venturi, and Julian Williams. FuturesMEX: secure, distributed futures market exchange. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 335–353. IEEE, 2018.

92 Nick Maxwell. Case studies of the use of privacy preserving analysis to tackle financial crime, Jan 2021. https://www.future-fis.com/uploads/3/7/9/4/3794525/ffis_innovation_

and_discussion_paper_-_case_studies_of_the_use_of_privacy_preserving_analysis_-_v.1.3.pdf.

93 Patrick McCorry, Siamak F. Shahandashti, and Feng Hao. A smart contract for boardroom voting with maximum voter privacy. In Aggelos Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 357–375. Springer, Heidelberg, April 2017.

94 Justice Ministry of Law and Company Affairs. The information technology act, 2000, 2000. https://bit.ly/3JuiUwy.

95 Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy*, pages 19–38. IEEE Computer Society Press, May 2017. doi:10.1109/SP.2017.12.

96 Ken Naganuma, Masayuki Yoshino, Hisayoshi Sato, Nishio Yamada, Takayuki Suzuki, and Noboru Kunihiro. Decentralized netting protocol over consortium blockchain. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pages 174–177. IEEE, 2018.

97 Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 129–139, 1999.

98 Neha Narula, Willy Vasquez, and Madars Virza. zkLedger: Privacy-Preserving Auditing for Distributed Ledgers. In Sujata Banerjee and Srinivasan Seshan, editors, *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, pages 65–80. USENIX Association, 2018. URL: https://www.usenix.org/conference/nsdi18/presentation/narula.

99 Chan Nam Ngo, Fabio Massacci, Florian Kerschbaum, and Julian Williams. Practical witness-key-agreement for blockchain-based dark pools financial trading. In *International Conference on Financial Cryptography and Data Security*, pages 579–598. Springer, 2021.

100 Christian Paquin. U-Prove Technology Overview V1.1. Tech report, Microsoft Corporation, April 2013. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/U-Prove20Technology20Overview20V1.120Revision202.pdf.

101 David C Parkes, Michael O Rabin, Stuart M Shieber, and Christopher Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294–312, 2008.

102 THE EUROPEAN PARLIAMENT and THE COUNCIL OF THE EUROPEAN UNION. Directive (EU) 2015/2366 of the european parliament and of the council, Nov 2015. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015L2366&from=EN.

103 Juha Partala, Tri Hong Nguyen, and Susanna Pirttikangas. Non-interactive zero-knowledge for blockchain: A survey. *IEEE Access*, 8:227945–227961, 2020. doi:10.1109/ACCESS.2020.3046025.

104 ARTICLE 29 DATA PROTECTION WORKING PARTY. Wp250: Guidelines on personal data breach notification under regulation 2016/679, 2018. URL: https://ec.europa.eu/newsroom/article29/items/612052/en.

105 Alexey Pertsev, Roman Semenov, and Roman Storm. Tornado Cash Privacy Solution, version 1.4, Dec. 2019. https://web.archive.org/web/20211026053443/https://tornado.cash/audits/TornadoCash_whitepaper_v1.4.pdf.

106 Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *21st ACM STOC*, pages 73–85. ACM Press, May 1989. doi:10.1145/73007.73014.

107 Peter Reuter and Edwin M. Truman. *Chasing Dirty Money: The Fight Against Money Laundering*. Peterson Institute for International Economics, 2004.

108 Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

109 Tomas Sander and Amnon Ta-Shma. Flow control: A new approach for anonymity control in electronic cash systems. In Matthew Franklin, editor, *FC'99*, volume 1648 of *LNCS*, pages 46–61. Springer, Heidelberg, February 1999.

110 Olivier Sanders. Efficient redactable signature and application to anonymous credentials. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 628–656. Springer, Heidelberg, May 2020. `doi:10.1007/978-3-030-45388-6_22`.

111 Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.

112 Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

113 Gerald Spindler, Anna Zsoofia Horvaath, and Lukas Dalby. Scalable oblivious data analytics: D.3.1 general legal aspects, 2017. URL: `https://soda-project.eu/wp-content/uploads/2018/02/SODA-D3.1-General-Legal-Aspects.pdf`.

114 Manu Sporny, Dave Longley, and David Chadwick. Verifiable credentials data mode, 2022. URL: `https://www.w3.org/TR/vc-data-model/`.

115 Manu Sporny, Dave Longley, Markus Sabadello, Drummond Reed, Orie Steele, and Christopher Allen. Decentralized identifiers (DIDs), 2022. URL: `https://www.w3.org/TR/did-core`.

116 Samuel Steffen, Benjamin Bichsel, Roger Baumgartner, and Martin Vechev. ZeeStar: Private Smart Contracts by Homomorphic Encryption and Zero-knowledge Proofs. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1543–1543. IEEE Computer Society, 2022.

117 Samuel Steffen, Benjamin Bichsel, Mario Gersbach, Noa Melchior, Petar Tsankov, and Martin Vechev. zkay: Specifying and enforcing data privacy in smart contracts. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 1759–1776, 2019.

118 Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. Mind your weight(s): A large-scale study on insufficient machine learning model protection in mobile apps. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 1955–1972. USENIX Association, August 2021.

119 Shahroz Tariq, Sowon Jeon, and Simon S. Woo. Am I a real or fake celebrity? evaluating face recognition and verification apis under deepfake impersonation attack. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 512–523. ACM, 2022. `doi:10.1145/3485447.3512212`.

120 Christopher Thorpe and David C Parkes. Cryptographic securities exchanges. In *International Conference on Financial Cryptography and Data Security*, pages 163–178. Springer, 2007.

121 Alin Tomescu, Adithya Bhat, Benny Applebaum, Ittai Abraham, Guy Gueta, Benny Pinkas, and Avishay Yanai. UTT: Decentralized ecash with accountable privacy. Cryptology ePrint Archive, Report 2022/452, 2022. `https://eprint.iacr.org/2022/452`.

122 Christof Ferreira Torres, Ramiro Camino, and Radu State. Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 1343–1359. USENIX Association, 2021. URL: `https://www.usenix.org/conference/usenixsecurity21/presentation/torres`.

123 Amos Treiber, Dirk Müllmann, Thomas Schneider, and Indra Spiecker genannt Döhmann. Data protection law and multi-party computation: Applications to information exchange between law enforcement agencies. Cryptology ePrint Archive, Report 2022/1242, 2022. `https://eprint.iacr.org/2022/1242`.

124 Marie Beth van Egmond, Thomas Rooijakkers, and Alex Sangers. Privacy-Preserving Collaborative Money Laundering Detection. *ERCIM News*, 2021(126), 2021. URL: `https://ercim-news.ercim.eu/en126/special/privacy-preserving-collaborative-money-laundering-detection`.

125 Xin Wang, Xiaomin Xu, Lance Feagan, Sheng Huang, Limei Jiao, and Wei Zhao. Inter-bank payment system on enterprise blockchain platform. In *2018 IEEE 11th international conference on cloud computing (CLOUD)*, pages 614–621. IEEE, 2018.

126 Sam M Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J Knottenbelt. Sok: Decentralized finance (defi). *arXiv preprint arXiv:2101.08778*, 2021. `https://arxiv.org/abs/2101.08778`.

127 Karl Wüst, Kari Kostiainen, Vedran Capkun, and Srdjan Capkun. PRCash: Fast, private and regulated transactions for digital currencies. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 158–178. Springer, Heidelberg, February 2019. `doi:10.1007/978-3-030-32101-7_11`.

128 Alex Luoyuan Xiong, Binyi Chen, Zhenfei Zhang, Benedikt Bünz, Ben Fisch, Fernando Krell, and Philippe Camacho. Veri-zexe: Decentralized private computation with universal setup. *Cryptology ePrint Archive*, 2022.

129 Arman Zand, James Orwell, and Eckhard Pfluegel. A Secure Framework for Anti-Money-Laundering using Machine Learning and Secret Sharing. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–7, 2020. `doi:10.1109/CyberSecurity49315.2020.9138889`.

130 Bingsheng Zhang, Roman Oliynykov, and Hamed Balogun. A treasury system for cryptocurrencies: Enabling better collaborative intelligence. In *NDSS 2019*. The Internet Society, February 2019.

131 Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town crier: An authenticated data feed for smart contracts. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 270–282. ACM Press, October 2016. `doi:10.1145/2976749.2978326`.

132 Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. DECO: Liberating web data using decentralized oracles for TLS. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1919–1938. ACM Press, November 2020. `doi:10.1145/3372297.3417239`.

133 Fairouz Zobiri, Mariana Gama, Svetla Nikova, and Geert Deconinck. A Privacy-Preserving Three-Step Demand Response Market Using Multi-Party Computation. In *13th Int. Conf. Innov. Smart Grid Technol.(ISGT North Am. 2022), Washingt. DC (to Appear)*, 2022. `https://www.esat.kuleuven.be/cosic/publications/article-3451.pdf`.

A PETs and the law

Usage of "classical" cryptography such as (public/private key) encryption, MACs, hash functions and digital signatures have been prevalent for decades and have so found their reasonable places within the legal framework of nations. For example legally speaking robust encryption schemes, used with long, high-entropy keys, provides a reasonable measure to secure sensitive data [62, § 83]. However, besides affirming the security provided by cryptography, the law can only be used to deteriorate the security cryptography offer. This can occur through the usage of back-door mandates, forced usage of insecure parameters or the forced assistance in by people and companies to break circumvent encryption as is for example the case in India [94, Sec. 69]. This is not something new, and goes back to 90's, where export restrictions of highly secure encryption schemes were in place in the US [59]. Even more concerning than imposed weaknesses in underlying cryptography, is the legal issue of forced decryption. While this has generally been an issue for individuals [46], it has also become an issue for companies. Apple was for example mandated to decrypt a user's iPhone the San Bernardino terror attack of 2015 [78]. Compelling a company to decrypt data should in itself not be regarded as a problem for PETs, as such cases already occur in non-private computation. However, this does become an issue in situations where PETs could be used to carry out transactions, which would otherwise not be possible due to their sensitive nature. In such a case, legal requests could also become preemptive requirements. Thus meaning that even if a company itself might not be trusted to not behave maliciously, it could be compelled to do so by its government. This is concretely a problem when entities in distrusting countries need to collaborate. One entity might trust another one to *not* behave maliciously, perhaps due to the public backlash of getting caught actively cheating[13] However if an entity acts malicious due to a governmental mandate, then such a backlash will be practically non-existing. Thus such legal potential could require the use of the strongest possible models of security.

While law could be used to break the security of PETS in certain weaker models, PETs can also help keeping the law. An example of this can be seen in the need for different governmental institutions or law enforcement agencies to share personal data in order to e.g. thwart terrorist plots. The need can be as simple as checking whether the same individuals are present in two different databases, and if so share relevant data. However, doing so without the aid of PETs would require leaking the individuals present in at least one of these databases. This goes against privacy laws, and in jurisdiction such as the EU, This would require explicit consent by all people in the database. While lawful sharing of personal data within law enforcement agencies might be possible in certain situations, such as sharing the list of publicly convicted criminals, this is not possible when it comes to other potentially relevant databases, such as people with a record of mental illness or people with gun permits. A study of how to use MPC in such cases has been thoroughly studied in both a cryptographic and legal framework by Treiber *et al.* where different law enforcement agencies aim to find and share data about common entries in their databases, under approval by a judge [123]. This framework could also prove relevant within law enforcement and financial institutions to ensure legality, such as law enforcement requests to the financial institution e.g. in relation to anti-money laundering (see Sec. 3.1). Thus PETs can be used to ensure the rights of citizens when the law gets involved. However, the law can also hinder certain computations

[13] Certain flavours of MPC [7] can for example be used to ensure that if someone tries to cheat, the honest parties will get a publicly audible proof of this, which can then be release to the appropriate authorities.

and model on certain types of data as we discuss in the following.

GDPR and MPC. On May 25th, 2018 the General Data Protection Regulation (GDPR) [62] came into effect in the EU. The law dictates how data concerning EU citizens, should be protected and handled, along with legal requirements concerning individuals' rights; such as requiring appropriate consent for data storage and computation, and the possibility of withdrawing such consent.

While the GDPR is an 88 page law document consisting of 11 chapters and a total of 99 articles, its general gist can be described from the a few terms:

Personal Data : Any information that relates to an individual, which can be directly or *indirectly* used to identify the individual. For example name, gender, social security number, religious belief, web cookies, etc. It should be noted that some pieces of data alone uniquely defined an individual, such as social security number, whereas as some pieces of data are just indirect identifiers, such as gender, town, religious belief. A single indirect identifier does not uniquely identify an individual, but combining several of these *may* uniquely identify the individual. The GDPR considers even a single indirect identifier as personal data.

Data processing : Any *process* performed on data. For example computing, storing, transmitting, deleting, etc.

Data subject : The legal person whose data is processed. For example a customer or a web page visitor.

Data controller : The entity who decides what actions will happen to personal data. I.e. the holder of such data. For example the employee at a company responsible for data storage. In practice a data controller will generally be consider a legal entity such as a company, who holds personal data. For example Google, Tesco, Die Bahn, etc.

Data processor : A third party that performs actions data on behalf of a data controller. For example cloud storage providers such as Amazon or Microsoft, or researchers computing statics.

Based on these definitions the GDPR requires that data gathering must be as *minimal* as needed and *measures* must be taken to ensure that personal data kept is up to date. Furthermore, any processing must be as *minimal* as needed, *transparent*, and done in a way that ensures *confidentiality* and *integrity.* Finally the data controller must be able to *demonstrate* compliance with the requirements. The GDPR has further requirements such as disclosure of breaches within 72 hours and in some cases the need for employee training and the appointment of a data protection officer. The requirements of a data controller is unsurprisingly much higher than for a data-processor. For example, a data controller requires consent from the data subjects for storage, computation and other actions on their data. These are not required by the data processor, although the data processor must still ensure that the data is protected and can still risk huge fines for failure to do so.

Relating the GDPR to PETs we see that the main question is whether any legal entity other than the data controller has access to personal data on its data subjects. This becomes a question of what "access" means. The GDPR states that if it is not *reasonably* possible recover the identity of persons based on the data in their possession, then they are not data controllers [113, 3.2.1]. Furthermore, there is precedence suggesting that a server simply storing encrypted data, for which it does not hold the key, is generally out of scope of GDPR requirements [104, Sec. II. D.]. Thus the legal requirements and scope of different entities participating using PETs on personal data comes down to the definition of *reasonably.*

For secure hardware is seems reasonable to assume that outsourcing the computation on personal data would either not be considered reasonable, *or* the provider of the secure hardware would be considered a data controller. E.g. for an execution on personal data on SGX on a server, either 1) the server and the initial data controller together would be considered a joint data controller (if SGX does not reasonably protect the data). Or 2) the initial data controller and Intel would be considered joint data controllers (as Intel is claiming their hardware reasonably secure).

In the case of secure computation, the situation becomes more unclear, firstly since it is undefined if collusion can be considered *reasonable*, and secondly since there is no single legal entity that can be held accountable in case of a breach. Thus there is a chance that the servers executing MPC would all be considered joint controllers on the set of all personal data which is computed on. This is called the *absolute* interpretation of the law. A less conservative view would imply that they become data processors, and thus need to adhere to general data safety and operational requirements, but do *not* need consent from the data subjects whose data they compute on, assuming they gave their data controller consent to carry out such a computation [3, 2.2]. This means that MPC could be used as tool for proving data privacy by design. However, the general consensus is that if data is secret share or encrypted when computed on, in a way where a single malicious legal entity is not able to *reasonably* recover the personal data, then the data is no longer personal and out of the scope of GDPR [113, 3.2.1] *if* the result of the computation does not *reasonably* allow deanonymization of the individual's whose personal data was used [81]. Crucially this applies, not just for a single computation but for the conjunction of all computations done on the data subject's personal data. This is known as the *relative* interpretation. Technically, the interpretation comes down to whether personal data is considered *anonymous* if no-one can reconstruct or if no single legal entity can reconstruct the data (up to cryptographic hardness).

More concretely, regardless the legal status of the different servers, when it comes to computing on personal data, the GDPR imposes other requirements in that 1) data subjects must consent to the *specific* computation to be carried out, at the time they give their personal data and 2) and the result of the computation must not be able to lead back to the data subject's personal data or identity. Thus according to the GDPR care must also be taken to only perform computations on personal data, where the result of the computation cannot lead to deidentifying the data subjects

One interesting exception to personal data under the GDPR is *pseudonomyzation.* Pseudonomization involves replacing identifiable parts of personal data with pseudonyms. It should not be confused with *anonymization*, which completely removes the coupling of personal data to a data subject. Pseudonomyzation is considered a reasonable approach to securing raw personal data, but at the same time is still considered personal data! Hence sharing and computation on pseudonomized data still make participating entities data processors at a minimum.

An example application of how to achieve utility from personal data without breaking privacy was demonstrated by Damgård *et al.* [51]. They constructed a scheme based on MPC where personal data held by a consultancy house, was used to compute credit scores (relative to other applicants in the same category) for load applicants, hence helping banks to give rational interest requirements and loan offers. Their concrete case was focused on Danish farmers, as they are not required to publish financial information about their business, and thus it is a challenge for banks to give reasonable loan requirements as they don't know how the specific applications compare to the general market. While not done explicitly the protocol of Damgård *et al.* could have allowed banks to give farmers loan offers without the farmers needing to disclose their financial situation to the bank beforehand.

Other laws. Many countries have privacy laws but few have been studied in the relation to PETs and to keep the scope of this survey simple we have only covered GDPR. Furthermore, the scope of GDPR is in a sense the one of the strictest privacy framework when comparing e.g. with HIPAA and CCPA in the US.

PETs assisting businesses and governments. Disregarding personal data, there are often other situations where companies, or even governmental instances might want to validate that the other party holds the information they claim they to do. This could for example be the case of mergers and acquisitions, where one company claims to hold some algorithms or machine learning model capable to perform certain actions, but they would of course not want to disclose this before the acquisition is complete.

A similar problem has also occurred during trials, such as the case of U.S. v. Michaud and U.S. v. Coplon, where the defendants were charged based on evidence gathered through the use of proprietary and secret law enforcement software. However the defendants defense wanted to inspect the software to ensure that it did not have faults and that it did not violate the Fourth Amendment (unreasonable searches and surveillance). In situations like these, zero-knowledge proofs could be constructed and used to convincingly validate the necessary constraints, without leaking proprietary information [19]. The same is true when the proprietary information is not a program, but classified data instead. If the data has been certified by an authority, then zero-knowledge proofs could be used to validate such data against a public predicate without leaking any content. Such zero-knowledge proofs could furthermore be augmented to allow for *public verifiability*, meaning that *any* external party can validate the proof. By combining this with a blockchain, such proofs could remain publicly accessible for anyone to validate. The possibility for public verifiability is for example highly relevant in the case of U.S. secret laws. These are laws whose content, or even existence, are classified. Such laws for example contain constraints on how law enforcement are allowed to circumvent security measurements to access people's personal data, without warrants. In court cases it is useful for the public and jury to be able to validate that the constraints in the secret laws have been obeyed by law enforcement [76].

In the finance sector such approaches could also be useful, for example for banks to prove that their AML software fulfill the minimum legal requirements, without leaking their proprietary algorithms, see Sec. 3.1. Such proofs could also be useful to post publicly for public verifiability, allowing (potential) customers to validate that their bank of choice is following legal requirements. Another example where this might prove useful, is in the situation of acquisition where information about the quantity, or demographic, of customers might be highly relevant to the price purchaser is willing to pay.

B Digital Asset Custody

As the vast majority of financial transactions are automatically executed over vast inter-bank and payment networks, the signing and encryption of sensitive transaction messages require secret cryptographic key material that must be carefully managed to prevent impersonation, theft and forgery. In traditional finance, this is implemented at the device instance level with hardware security modules (HSM). These generate, store and use sensitive key material locally; any signing or encryption operations are performed strictly on the HSM such that the key material never leaves the device during usage. Thus, physical access to HSMs must carefully guarded and its operation must adhere to rigorous standards, such as those set by the Payment Card Industry (PCI).

HSMs are widely used in payment networks, for example, when the EMV protocol [11] for credit card transactions requires the online verification of a customer PIN entered at the point-of-sales (POS). In this case, the PIN must be forwarded in authenticated and encrypted form to the card issuer for verification. However, as the POS supplier does not have a direct relationship with the card issuer, the PIN is forwarded via hops between intermediaries, where only neighboring intermediaries have established shared keys for encrypted communication; Such keys for encryption are managed by HSM's. However, security which HSM provide also means they are costly to acquire (tens of thousands of dollars) and to operate, as even the most basic maintenance items such as firmware upgrades must be conducted on-site and involve multiple, redundant operators with requiring specific access authorizations. Furthermore, strict PCI standards on the design and operation of HSM's make changes to the underlying cryptography very difficult, as custom HSM firmware are generally not permitted.

Threshold signatures computed by MPC committees have emerged as an alternative to HSMs in the context of digital assets, which have deployed non-standardized signature schemes such as ECDSA based on non-standard elliptic curves (e.g. secp256k1 in Bitcoin). Whilst HSMs have traditionally played the role of secure signing, they are difficult to customize and adapt in an industry with rapidly evolving cryptography on ever newer blockchain protocols. Furthermore, since HSMs represent a single point of failure, their installation and physical access control requires expertise, not readily available to fast-moving cryptocurrency startups and institutions. MPC can be offered as Software-as-a-Service or cloud solutions. Suppliers of such digital asset custody solutions include Unbound[14] [47] and Fireblocks [63], which have implemented highly performant threshold signing algorithms [88, 34, 53]. In contrast to HSMs, MPC servers can be run on standard virtual cloud instances, offering a high level of elasticity for both (1) performance and (2) key security; (1) Signing of independent transactions to authorize transfer of digital assets is easily parallelizable and (2) the number of MPC servers can be scaled to increase decentralization of the key material. MPC instances can also be easily upgraded to perform new cryptographic tasks as there is a lesser need for standardization and hardening of individual MPC devices given a lack of a single point of failure. Furthermore, using MPC allows for easy and simple portability and back-up of key material. Another benefit of using this technology is employing the concept of *proactive security* for MPC protocols to periodically refresh their internal scret states in order to recover from momentary security compromises. In particular, this has been implemented in the context of MPC-based key management for digital asset custody [34].

We note several critical differences between traditional finance and decentralized finance, which explain the quicker adoption of MPC by the latter. In traditional finance, transactions can generally be revoked; if an individual HSM is compromised and its key material exposed, the log of the attack can be traced to an individual device. The post-mortem analysis can thus establish a clear breach event and entry point, providing clear evidence that a transaction was authorized with stolen keys for its later revocation. In contrast, transactions in digital ledgers can generally not be reverted; they are final. For this reason, cryptocurrency exchanges and custodial services will operate "cold" and "hot" wallets; the former are generally disconnected from the business logic and require human authorization to move funds. The latter hold a fraction of the total assets, but are automatically triggered to produce valid digital signatures when prompted by the customer-facing application.

We consider the adoption of MPC for the application of digital asset custody an illustrative one; the improved updateability and tuneability of both performance and security in MPC

[14] Acquired by Coinbase, Inc.

(over classical HSM's) is a major selling point for emerging applications, and we anticipate MPC to spread to other key management domains in finance over time.

C Privacy-Preserving, Demand Response Markets

Privacy in markets is applicable to other domains, such as demand-response auctions; the growth of sustainable energy generation has introduced the necessity of increased coordination between the *production* and *consumption* of electricity. In contrast to traditional power generation sources, such as gas turbines, which can be throttled to match current power demand on the grid, sustainable power sources such as wind or solar cannot. Today, grid operators purchase future *flexible demand* from large, industrial scale consumers of energy or even power generators with flexibility to scale production in either direction to balance the grid.

There are efforts to aggregate retail consumers of electricity to form sellers of demand-response capacity; for example, home HVAC system of buildings can easily shift their power consumption forwards or backwards in time whilst maintaining set temperature. Residential electric vehicle charging stations can easily defer charging until opportune time periods to stabilize the grid, without compromising the convenience of the owner. However, serious privacy concerns arise when submitting device-level power consumption constraints to a demand-response auction run by the utility, as such information easily reveals the type of activity occurring in private homes.

Thus, in the work of Zobiri *et al.* [133], future demand capacity at the retail device-level can be sold to buyers; in this work, each auction round for a time frame within the *day-ahead* will accept submitted bids that include the maximum power consumption or power draw (KW), price limit ($/KWh) as well as characterization of each the demand-flexibility of each individually controllable device. For example, a washing machine can only provide demand flexibility for its starting time; once activated, a washing cycle cannot be interrupted. In contrast, an electric vehicle charger can charge intermittently at different power levels, but must completely charge the vehicle by a certain time. Thus, the auction does not simply involve buy and sell bids by "volume" and "price"; rather, it must take into consideration the forecast, future consumption-demand of the buyer, and device-level constraints submitted by sellers. When implemented in MPC, this arguably leads to a more complicated auction algorithm than in traditional markets; the work [133] permits private bids to be submitted to capture the consumption flexibility as *constraints* for each device (temporal constraints, power consumption cycle constraints), in addition to a electricity price *limit*; power consumption constraints are aggregated over private bids inside the MPC, such that device-level information is protected from the buyer of demand-response capacity. The auction clearing mechanism must then match device-level consumption constraints against predicted power generation schedule published by the buyer. Thus, demand-response markets imply direct device scheduling in residential homes by the buyer (or utility operator); privacy of end-users must therefore be protected by such solutions where any scheduling information is evaluated inside an MPC instance.

Da Gama *et al.* [67] propose a peer-2-peer electricity market run a similar MPC setting, where local producers and consumers of electricity can trade energy *intra-day*; here, the auction design resembles that of the author's prior work in dark-pools [48], and exhibits sufficient transaction throughput for intra-day auction applications. [67] builds on prior work [1], but also considers geographical proximity of buyers and sellers, thereby stabilizing grid operations as power generation and consumption can be optimized to occur locally.