panther.

WHITEPAPER

Privacy preserving protocol for digital assets

Version 1.0.1 | 19th July, 2021

PantherProtocol.io

Panther Protocol Whitepaper

The Panther Protocol Team

Version 1.0.1

19th July, 2021

Panther is a decentralized privacy meta-protocol enabling **confidential, trusted transactions** and **interoperability** with DeFi.

This whitepaper presents Panther Protocol, a privacy protocol for digital assets. The Panther Protocol's main goals are:

1. providing a secure, private transaction ecosystem with a superior user experience,
2. maintaining composability with decentralized finance (DeFi) protocols,
3. ensuring privacy backed by a game-theoretic model,
4. establishing verifiable trust relationships between participants, and
5. developing a novel price discovery mechanism for privacy.

This whitepaper aims to weave together the topics of DeFi, privacy, trust, compliance, economics, game theory, and privacy-enhancing technology (such as zero-knowledge proofs) into a coherent thesis that motivates and inform the design of the Panther Protocol. We will dive into Panther's architecture, technical design, governance, and tokenomics, and finally, conclude with a reflection on where Panther is now and where it will take us in the future. This is a "living" whitepaper that will be updated as our work and community develop over the course of the project.

Contents

List of Figures

List of Tables

1 Introduction

Today, decentralized finance (DeFi)[1] applications are predominantly built on the Ethereum protocol where all transaction history and balances are public by default. In fact, most blockchains lack privacy protection, making it difficult for investors and users to conduct trades freely and confidentially.

Whilst privacy-native cryptocurrencies exist (e.g. Zcash, Monero), they are not composable with the DeFi ecosystem, limiting their utility. Separately, privacy protocols that exist today were not designed with regulatory compliance in mind. This discourages institutional investors from experimenting in privacy protocols, which creates a huge challenge in bootstrapping liquidity and privacy, preventing private assets from becoming mainstream.

Panther is an end-to-end privacy protocol for digital assets, which can be deployed in a compliant way on any public blockchain.

Key differentiators / features:

- Enables trusted transactions and regulatory compliance whilst preserving privacy
- Privacy mining and price discovery mechanism
- DeFi composable
- Interoperability between multiple blockchains

We provide in Table 1 a high level comparison between the design of Panther and the status quo of several other privacy-preserving protocols.

In this paper, we first set the scene by providing a brief overview of the current decentralized finance landscape. We move on to the subject of privacy, examining the topic from ethical, economical, and technical standpoints, and how it applies to the world of DeFi. We then describe the Panther Protocol in detail in terms of its development, technical design, governance, and token economics. We end with some concluding remarks on what we hope to achieve, and consider future directions for this project.

Project	User Experience	Trust / Compliance	DeFi Composable	Game-theoretic security	Cross-chain
Zcash	Average	×	×	×	×
Tornado Cash	Complex	×	×	×	×
zk.money	Good	×	WIP[2]	×	×
Incognito	Good	×	WIP[3]	×	✓
Panther[4]	Good	✓	✓	✓	✓

Table 1: Comparison between Panther and other privacy protocols.

[1] For an introduction to DeFi please refer to [30]

[2] At the time of writing, Aztec had recently announced they are in the early stages of development of interoperability with DeFi protocols.

[3] See Incognito's 2021 roadmap.

[4] Since Panther is a new project, this row reflects Panther's design and key focus areas.

2 Background on Privacy

For millennia since the dawn of civilization, all human communications have been local and ephemeral, and therefore necessarily private. From ancient Rome to the modern day, as the speed of communication and data processing capacity grew exponentially over the centuries, so did the temptation, means and opportunity for those in power to eavesdrop on conversations and collect personal data on a grand scale. The privacy landscape is therefore inextricably linked with the evolution of technology.[5]

In this section we examine how the blockchain technology and modern mathematics can shape the privacy landscape in the era of decentralization and advanced cryptography.

2.1 Privacy on Blockchains

Blockchain technology is celebrated for paving the way to decentralization by removing censorship and anti-competitive power from traditional centralized institutions. Paradoxically, however, while blockchains take away control from institutions, their openness makes them a perfect platform for targeted monitoring as well as mass surveillance.

In order to achieve network consensus, the conventional blockchain model requires total transaction transparency. That is to say, all transaction data, including sender and recipient addresses, value, currency or token type, smart contract data, and transaction timing are all in the clear for all to see. Blockchains are thus a treasure trove of data representing user's private remittances, financial holdings, Non-Fungible Token (NFT) purchases, currency exchanges, etc. These can often be supplemented by off-chain metadata to unmask the real world identity behind the wallet addresses.

In a research note [3], Angeris et al. found that in the context of a type of decentralized exchange known as a Constant Function Market Maker (CFMM), "privacy is impossible with the usual implementations of CFMMs under most reasonable models of an adversary and provide some mitigating strategies".

Fortunately, advances in cryptography and privacy-enhancing technology in recent years have allowed us to claw back much-needed privacy from blockchain transactions. For further reading into the subject, we refer the reader to the survey by Bernabe et al. [5] as well as the list of papers and articles found on [23].

2.1.1 Notions of Privacy

Here we define some terms related to notions of privacy properties on blockchains that we may use in later sections.

Unlinkability: A set of transactions is unlinkable if a third-party observer is unable to determine (with a level of confidence) whether any sender's or recipient's identity belongs to the same individual, and therefore unable to backtrace the origin of any of the transactions.

Third-party anonymity/privacy: A transaction is third-party anonymous (or private) if the

[5] For the reader interested in the subject of privacy, Chapter 26 of [2] provides a general discussion of surveillance and privacy, while *surveillance capitalism* is explored in [34] and a history of wiretapping in [8].

sender and recipient can identify each other but a third-party observer cannot learn anything about either party.

Private transaction: A transaction that is fully obscured, meaning all "interesting" data fields, including sender, recipient, currency, amount, and fees cannot be extracted by a third-party observer. The existence of the transaction itself remains observable.

Unobservability: A transaction is unobservable when the timing and the existence of the transaction itself cannot be determined by a third-party observer.

2.2 Financial Privacy vs Compliance

2.2.1 Financial Regulations

One of the most influential organizations in combating financial crimes is the Financial Action Task Force (FATF), an inter-governmental institution founded in Paris. Its official recommendations are ratified by its member states into its own laws.

Under FATF recommendations, the financial institutions are required to comply with Anti-Money Laundering/Countering the Financing of Terrorism (AML/CTF) regulations. Compliance with such regulations typically involves customer due diligence and risk assessment, transaction monitoring, and fulfilling a wide range of reporting to the authorities.

In the cryptoassets space, Financial Crimes Enforcement Network (FinCEN) is one of the major bodies that regulates activities on blockchains in the United States. In a recent controversial proposal, FinCEN proposed a new rule[6] for cryptocurrency exchanges to have to maintain customer identity information on transactions over $3,000 and submit to them reports with the same details if over $10,000.

2.2.2 Problems with Compliance

In the developed countries, the current compliance regime operates under something akin to a *guilty unless proven innocent* presumption. Segments of society are denied access to financial services. Customers' Personally Identifiable Information (PII) and transaction data are by default collected, stored, data-mined for patterns and subject to sharing with third parties and authorities. A summary of benefits and drawbacks in the current financial system is illustrated in Figure 1. At any time, financial assets may be frozen, confiscated, devaluated, or undergo a "hair cut", sometimes justifiably, often not.

We outline below some of the shortcomings of the current financial compliance regime.

Privacy. Customers of financial services are typically subjected to lengthy and intrusive scrutiny into their identity and personal history. Once on-boarded, they are subject to continuous monitoring of transactions that will reveal their sexual orientation, health conditions, political preferences, places traveled, social interactions, associations and so on. In the wrong hands, such information may be used for blackmail, denying insurance, stalking, denying job applications, inclusion on the no-fly list, etc.

Ineffectiveness. There is little research on the efficacy of AML/KYC regulation. Regula-

[6] See https://home.treasury.gov/news/press-releases/sm1216

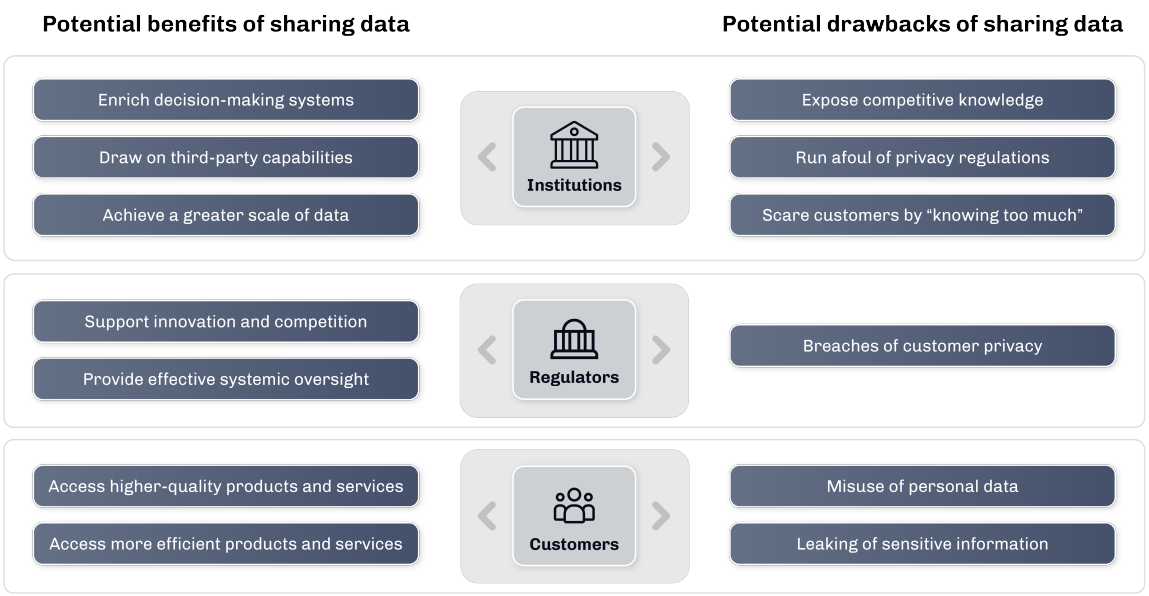

Figure 1: Data Sharing Benefits and Drawbacks (WEF/Deloitte)

tors themselves appear only interested in imposing ever more stringent requirements without pausing to evaluate whether the measures achieve their objectives, and more importantly whether they justify the monetary and privacy costs.

A recent study by Pol [21] finds that "the anti-money laundering policy intervention has less than 0.1% impact on criminal finances, compliance costs exceed recovered criminal funds more than a hundred times over, and banks, taxpayers and ordinary citizens are penalized more than criminal enterprises".

Separately a report by the British Banking Association (BBA) mentions "The costs of financial crime compliance for the British banking industry have gone beyond a 'tipping point' in comparison with the AML/CTF benefits being accrued".

Regulatory conflicts. Often contradictory to data privacy legislation such as the EU's GDPR. One example is data sharing between EU and non-EU countries. The unharmonized regulatory frameworks make it risky for financial services to fall foul of one set of regulations when trying to comply with another.

Inefficiency. Users interacting with multiple institutions must repeat essentially the same KYC procedure with each one separately. There is not an infrastructure for reusable KYC proof that a user can be presented to multiple verifying organizations. These duplicated efforts impose costs in terms of time and monetary, as well as in increased risk of data breaches [17].

Monetary cost. Compliance is costly to implement and manage. According to a study [27] the AML compliance costs for US and Canadian financial institutions for the year 2019 at over $31.5B, at an average of about $1.5M for small firms and over $14M for mid/large firms.

Security risks. Collection and warehousing of PII and transaction history is a liability, some might even describe it as a toxic asset [25]. Threats include abuse of data, unauthorized access by staff, data breach by malicious actors (including nation-state). High costs for implementing protection against these threats in terms of policies and procedures, authentication

and authorization systems, continuous monitoring, auditing. Still, breaches continue to happen. To illustrate the scale of the issue, breaches in China's Sina Weibo and Marriot International in the US, two of the most high profile breaches in recent years, have resulted in compromise of over one billion user accounts.

Erosion of trust. A survey commissioned by IBM [31] shows that only 20% of US consumers "completely trust" the organizations they interact with to maintain the privacy of their data. Customers fear that their data could be used to harm them (e.g. through identity theft) and more broadly that unintended parties can learn something about them that they wish to keep private (e.g. sensitive purchase history). This is hardly surprising given some of the recent high profile cases affecting a large portion of US Citizens, including data abuse scandal at Cambridge Analytica and breaches at Capital One, Experian and Equifax.

Financial exclusion. According to Jonathan Fisher QC in [22] "On any view, the anti-money laundering and counter-terrorist financing regime aggravates financial exclusion by systematically excluding certain groups of people and businesses from products and services offered by financial institutions. The significance of the problem cannot be underestimated, since financial exclusion can have a devastating impact on individual lives, the business community, and society in general." The impact includes exacerbation of inequality, reduced access to consumer protections, stunted innovation and reduced levels of overall economic development.

Indeed, FATF has recently officially recognized financial exclusion as an "unintended consequence" of their standards and has launched a new project to address it [12].

Non-universal and unjust. Different rules apply for governments and powerful corporations vs ordinary citizens. Governments can finance rebel/terror groups to bring about regime change. Corporations and high net worth individuals can set up complex tax evasion schemes.

Anti-competitive. Smaller firms are faced with more challenges with AML compliance [27] than larger firms. This creates a significant barrier to entry for start-ups and thus deters financial innovations.

2.2.3 Solutions

While it is not our aim to address every shortcoming with the current compliance landscape (nor can we hope to do so), we wish to at least partially redress the compliance cost/benefit balance by advocating the use of Privacy Enhancing Technology (PET). In addition to harmonizing with privacy regulations, PET also has the potential to increase efficiency, lower costs, and enhance the security and trustworthiness of the financial system.

The problem with traditional compliance stems from the assumption that in order to detect financial crime, it is necessary to gather and analyze large amounts of raw data. However, with the advancement of computer science and mathematics, this assumption no longer holds true.

Recent research and development into PET has allowed a party to verify, process, and make inferences from a target data set, without directly accessing the underlying data. PET is thus a practical manifestation of the Least Privilege Principle, a cornerstone of sound system design for secure, dependable systems.

We provide a brief survey below. Some of the PETs described here will be further explored in

a later section.

Differential Privacy: A family of techniques which add noise to a data set so that it is impossible to reverse-engineer the individual inputs [10].

Homomorphic Encryption: Special encryption schemes the output of which can be shared with a potentially untrusted third-party for computation and analysis, but not decryption back into the original data [32].

Zero Knowledge Proofs: Methods with which users can prove their knowledge of a value without revealing the value itself [14].

Secure Multiparty Computation: Protocols with which data computation is spread across multiple parties such that no individual party can see the complete set of inputs [11].

Selective Disclosure Schemes: Also known as Anonymous Credentials Schemes, these are proof-of-knowledge schemes in which the prover who possesses some certified credentials containing multiple attributes may, at their discretion, choose to disclose a selected subset of those attributes to a verifier.

2.2.4 Can Privacy and Compliance Co-exist?

While we do not presume all the above issues can be practically and immediately addressed by PET, regulators should recognize the development of PET and its potential. They should start, at the very least, to entertain the possibility of embracing PET and move away from the current prescriptive modus operandi, that of non-discriminatory wholesale data collection, analysis, and sharing.

More generally, they should take a more scientific and evidence-based approach in policy-making, taking into account costs both tangible and intangible (such as privacy), actual realized benefits (crime reduction), and technological solutions that have the potential to achieve their goals in a far more effective and dignity-respecting manner. We believe that in the long term this will lead to a global financial system that is far more inclusive and just. Those to embrace this approach can deservedly claim the moral high ground over regimes that do not respect privacy and other forms of basic human rights.

3 Economics of Privacy

In this section, we will discuss pure economic aspects of privacy, especially from a microeconomic perspective. Empirical research suggests that most people usually place a low value on their privacy.

Often, however, it is not clear why. Is this a deliberate choice of an individual, or insufficient awareness of the risks? The latter is present in the misunderstanding of how much economically relevant information is shared about people on certain platforms.

Developing one unifying economic theory of privacy is impossible because economically relevant privacy issues arise in many different contexts. There are diverse situations where the protection of privacy can both increase and decrease individual or social welfare. In modern digital economies, consumers cannot make informed decisions about their privacy, because

consumers often have incomplete information regarding when their data is collected, how it is used, and what the consequences of this use are.

Privacy means different things to different people. We constantly navigate privacy boundaries, both as individuals and as consumers. The decisions we make about them determine explicit and implicit benefits and costs, both for ourselves and for society.

Particular interest in privacy is focused on its informational aspect: the trade-offs from protecting or sharing personal data. Some sub-fields of information economics are related to privacy economics, because they deal with the trade-offs coming from the public or private states of information. For example, an auction can be designed so that participants reveal their true costs or valuations, or taxes can be designed in such a way that it is optimal to report truthfully. Research on auctions and optimal taxation deals with the private information of economic agents (e.g., consumers, firms), while privacy economics deals more specifically with the personal information of actual individuals.

One of the basic examples of individual economic losses caused by not having privacy is the following. Suppose there is a buyer, and a seller, who owns an item. Suppose the valuation of the buyer for the item, his reserve price, is v_B. The valuation of the seller for the item, his reserve price is v_S. Both are private information. There are gains from trade if and only if $v_S < v_B$, and gains are equal to $v_B - v_S$. The main question of commerce is: how are these gains shared? To obtain these gains, the price p, so that $v_B < p < v_S$, has to be found. This question is studied under the name of the *double auction* in economic literature. Finding optimal posted price p is a topic of [6]. Optimal price requires distributional assumptions on v_S and v_B. However, if the seller knows buyer-relevant transactions, he can estimate v_B closely and post a price p very close to v_B, therefore, taking all gains from the trade himself. On the other hand, if the buyer knows seller-relevant information, he can post a price p arbitrarily close to v_S and realize almost all gains.

Privacy of transactions is important in auction settings as well. Sealed bid auctions are omnipresent. In blockchain environments, however, often both the bids[7], and sometimes deposits are public information, which gives rise to undesirable behavior of the auction participants. The latter is the case, for example, in the namebase auctions. The possibility of observing deposits fundamentally changes the implications of the auction, especially if bidding happens sequentially. Since bidding is costly, the deposit has to be staked for a substantial amount of time and these can be used as a costly signal for a high valuation; this implies multiple inefficiencies. In order to engage in costly signalling, a bidder who bids first and has a high valuation, deposits a lot. If high valuations are likely, entry deterrence may happen through high deposits: a bidder who bids first can deter subsequent bidders from participating in the auction. Partial pooling may also happen in the equilibrium. Bidders of different valuations may deposit the same amount. The auction fails to allocate the item to the bidder with the highest valuation, which is the ultimate goal of the original protocol. See [24] for a formal treatment of implications.

Privacy of information, however, is not the opposite of sharing of information, rather, it can be used by a strategic player as a tool to optimize his payoffs. It can be seen as a control mechanism over sharing. Releasing information selectively can increase advantage in economic interactions, and therefore, expected utility.

[7] See for example Gnosis Auction.

A survey of various sub-fields in the economics of privacy is given in [1].

One notable example where the lack of privacy in the blockchain context causes loss of value is so-called Miner Extractable Value (MEV), sometimes referred to as Maximal Extractable Value. Consider the scenario where users place orders in an Automated Market Maker (AMM) decentralized exchange. Orders are of a type (a, b), where a denotes the amount of the first cryptocurrency that the user is spending, and b is the minimum amount of the second cryptocurrency the user wants to receive in return. Independently from the rule by which a particular AMM determines price, if the user is not aware of the true price relation between these two cryptocurrencies, there is a potential *sandwich* attack. The miner can place two orders, one before and one after the proposed order of the user, so that he/she extracts value by arbitrage. These types of attacks are characterized in [4].

4 Building Blocks for Privacy-Preserving DeFi

In this section, we provide a brief survey on the current mathematical and technological building blocks that allow us to develop practical privacy-preserving DeFi solutions.

4.1 Zero Knowledge Proofs

In cryptography, a zero-knowledge proof or zero-knowledge protocol (ZKP) is a class of method by which one party (the prover) can prove to another party (the verifier) that they *know* a value x, without revealing x itself or any other information.

The seminal work on ZKP was published by Goldwasser, Micali and Rackoff in 1985 [14]. While their proposed zero-knowledge scheme was not practical, the result demonstrated that ZKP was a mathematical possibility that sparked its continual research to this day. A zero-knowledge proof must satisfy three properties:

- **Soundness:** if the statement is false, the verifier will always reject.[8]
- **Completeness:** if the statement is true, the verifier will always accept it.
- **Zero-knowledge:** the verifier learns no information except for the truth of the statement.

Interactive ZKPs require interactions between the prover and the verifier when validating the proof, whereas non-interactive zero-knowledge (NIZK) proofs allow the prover to generate and publish a proof that can be validated by any verifier at any time with no further interaction. For this reason, non-interactive ZKPs are particularly useful in the blockchain setting.

Succinct Non-interactive Argument of Knowledge (SNARK) is a class of practical proofs which possesses the following properties:

- **Succinct:** the size of the proof is small compared to the size of the statement being proved.
- **Non-interactive:** it does not require rounds of interaction between the prover and verifier.

[8] Except for a negligibly small probability.

- **Argument:** a weaker notion of a mathematical *proof* where we assume the prover has bounded computational resources.
- **Knowledge:** the prover cannot construct a proof without knowing a certain *witness* for the statement.

A SNARK is not necessarily *zero-knowledge*. If a SNARK allows proofs to be conducted without revealing the witness, we call it a zero-knowledge SNARK or commonly zkSNARK. Generating a zkSNARK proof is a multi-stage process, an example of which is illustrated in Figure 2. For more details, an introduction to zkSNARKs and their recent development is presented in [19].

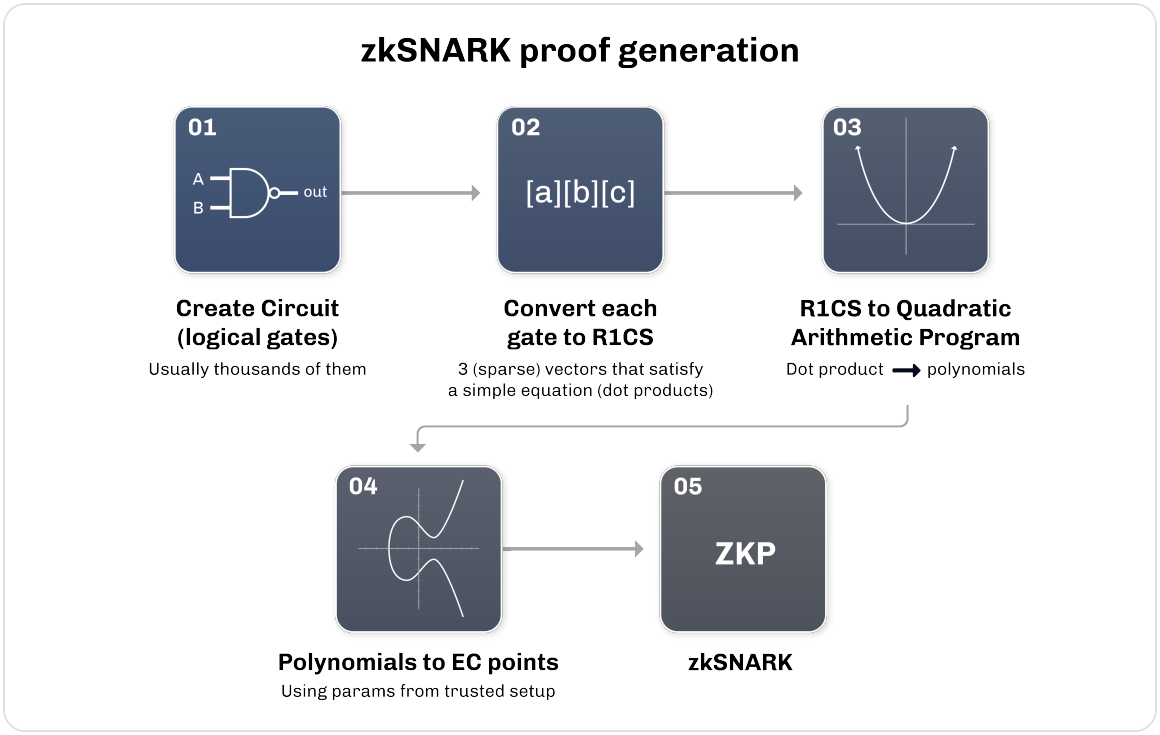

Figure 2: zkSNARK workflow

In DeFi and blockchain in general, zero-knowledge proofs are a solution to two different problems. Their *zero-knowledge* property provides **privacy and anonymity** to users' transactions and their utility as *proof of computation* are exploited to implement blockchain **scaling solutions**.

4.2 Secure Multi-Party Computation

Secure Multi-party Computation (abbreviated as *MPC*) is a subfield of cryptography with the goal of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private.

In an MPC, where we have $N > 1$ participants each have private input x_i, respectively $x_1, x_2, \ldots, x_N$. Participants want to compute the value of a public function $\mathcal{F}$ on that private data $\mathcal{F}(x_1, x_2, \ldots, x_N)$ while keeping their own inputs secret.

Yao [33] introduced the notion of MPC, exemplified in the hypothetical *millionaire's problem*:

'Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each other's wealth." In this instance $\mathcal{F}$ is simply $(x_1 \leq x_2)$ which outputs a Boolean result (true or false) and x_1, x_2 are the respective wealth of millionaires 1 and 2.

Secret Sharing (SS) is an important primitive in MPC protocols. As its name suggests, SS allows a secret to be split and shared among different parties. A useful property of SS is that we can set it up so that the secret is shared between n parties, k of which, where $(1 < k \leq n)$, must be combined to reconstruct the original secret. On their own, or in groups of less than k, the shares will reveal no information whatsoever about the secret.

For further study, a comprehensive treatise on MPC is available from [11].

4.3 Trusted Computing Solutions

In our context, Trusted Computing refers to the general concept of the use of an isolated computing resource which offers a measure of guarantee on the security and integrity of the computation even if the main application has been compromised.

Hardware Security Modules (HSMs) are physical devices commonly used in banks to provide secure, tamper-resistance storage of keys and their management. Typically, keys secured with an HSM never leave the confines of the device. Access to cryptographic operations is provided by application programming interfaces (APIs) which accept a *key handle* as a parameter. HSMs may also implement a security mechanism which wipes the key material if a physical attack attempt has been detected. Some HSMs offer programmability to allow custom code to be run inside the HSM.

A *Trusted Execution Environment* (TEE), also known as a *Secure Enclave*, is a secure area within a general purpose processor which guarantees confidentiality and integrity of code and data being executed within. In practice, this protects sensitive data from being accessed even if the main operating system is compromised. Some TEE implementations also support *remote attestation* which cryptographically proves that you are interacting with a genuine TEE (rather than a non-secure processor running the same code). Two popular implementations of TEEs are ARM's TrustZone on mobile devices and Intel's SGX on servers.

Confidential Computing takes the concept of TEE a step further by securing complete virtual machines (VMs), which among other things support real-time data encryption. For example, the Google Cloud Platform offers Confidential VMs based on AMD EPYC processors. Similar features are supported in OpenStack [29], the most widely deployed open-source cloud infrastructure globally[9].

By offloading sensitive computing workloads to a trusted computing environment, applications can provide and attest to a level of guarantee that private data cannot be accessed and used in an unauthorized manner. The Signal messenger application, for example, uses Intel SGX to allow a user to discover contacts already registered on the platform without exposing the contact list to the Signal employees [18].

[9] Due to an engineering effort previously led by a member of the Panther team [28].

4.4 Mixing Services

A mixing service (sometimes known as a *tumbler*) is a service that obscures the audit trail back to a cryptoasset's original source. This is usually achieved by breaking the link between the fund's input address and output address, thereby allowing an identifiable or "tainted" asset to become anonymous and untainted.

This in turn enables assets passing through the mixer to regain (or retain) their fungibility, which is crucial characteristic of sound money. If the fungibility of an asset class is not safeguarded, it cannot serve as a reliable and stable store of value, and therefore cannot be regarded as a hard currency.

For background, Chaum introduced the notion of providing privacy using mixing methods in [7]. A survey of techniques used by mixing services is provided in [16].

Loopix [20] is a recently proposed mixing-based anonymity system. It uses a mixing technique that is based on the independent delaying of messages, which makes the timings of packets unlinkable. Moreover, Loopix introduces a number of types of decoy traffic to thwart de-anonymization attacks.

The two main factors that affect the level of privacy offered by mixing services are:

- the size of the anonymity set,
- the volume of transactions in the anonymous pool prior to withdrawal, often proxied by *time in the pool*.

The larger these two parameters are, the harder it is for a third-party observers to track the flow of the anonymized cryptoassets, and conversely, the higher the level of *plausible deniability* a user can claim about their transaction activities.

4.5 Limitations of Privacy Enhancing Technology

While a casual understanding of PET may lead the reader to celebrate that the privacy problem is "solved", a deeper examination of the subject matter will point to a less Utopian, but still optimistic, conclusion.

At the core, the realization is that PET *shifts* trust rather than removing it from systems altogether. Whereas traditionally privacy depends on trust in the regulatory frameworks and organizational policies and procedures, PETs require trust of mathematical proofs and their interpretation, the strength of hardness assumptions of mathematical problems[10], protocol implementation correctness, the trustworthiness of software dependencies and the security of the computing resource supply chain.

In addition, the level of privacy and anonymity may be compromised by user error or limited by the small pool of participants (the *anonymity set*).

We are nonetheless optimistic as they are steps in the right direction of removing or at least reducing data available to and trust bestowed upon traditional organizations. We have seen areas of cryptography that have matured over the past few decades, for example, public-key cryptography was not so long ago a completely new concept. Today public-key based

[10] Intuitively these are what make cryptographic schemes difficult to break.

key agreement and signature algorithms form the backbone of Internet security and indeed blockchain technology itself.

Researchers have made tremendous progress in privacy techniques and there is every reason to expect the trend to continue so that one day we will expect zero-knowledge to be part and parcel of services that we receive, much like we expect seat belts and ABS brakes in passenger vehicles and authenticated encryption when browsing the web.

5 Implementation Options for Private DeFi

Currently, most programmatic functionality on blockchains is executed in smart contracts for which Ethereum is *the* dominant platform. The following solutions present the prevailing technical options for implementing DeFi privacy on Ethereum and other networks which do not preserve privacy.

5.1 Smart Contract Solutions

In smart contract based solutions, privacy is achieved by operations performed in the smart contract layer. The user deposits assets into a smart contract which in turn performs operations such as mixing. At the end of the operation, the smart contract makes the anonymized asset available to the user to be withdrawn to a fresh, unused address.

Figure 3 illustrates a mixer solution implemented using smart contracts.

5.2 Private Cross-chain Solutions

Figure 4 illustrates a privacy solution which enables DeFi operations to be handled by a non-privacy-preserving Layer 1 chain (Ethereum) on behalf of a privacy-preserving chain. This approach requires a custom adapter to be created for each of the applications with which it needs to interact, and these communicate across bridges to the Layer 1 which handles the DeFi transactions.

5.3 Private Layer 2 Solutions

Layer 2 solutions provide privacy by mixing transactions off-chain before committing on chain, through the use of e.g. ZK-Rollups[11]. Besides processing users' balances privately, this often allows many transactions to be bundled together off-chain which are periodically synchronized onto the blockchain in a highly compressed form. This approach increases transaction throughput and provides savings in transaction fees such as gas costs when operating on top of Ethereum, although it is likely to result in increased transaction latency.

A survey of Layer 2 privacy enhancing solutions is provided in [15].

[11] An introduction to rollup solutions can be found in [13].

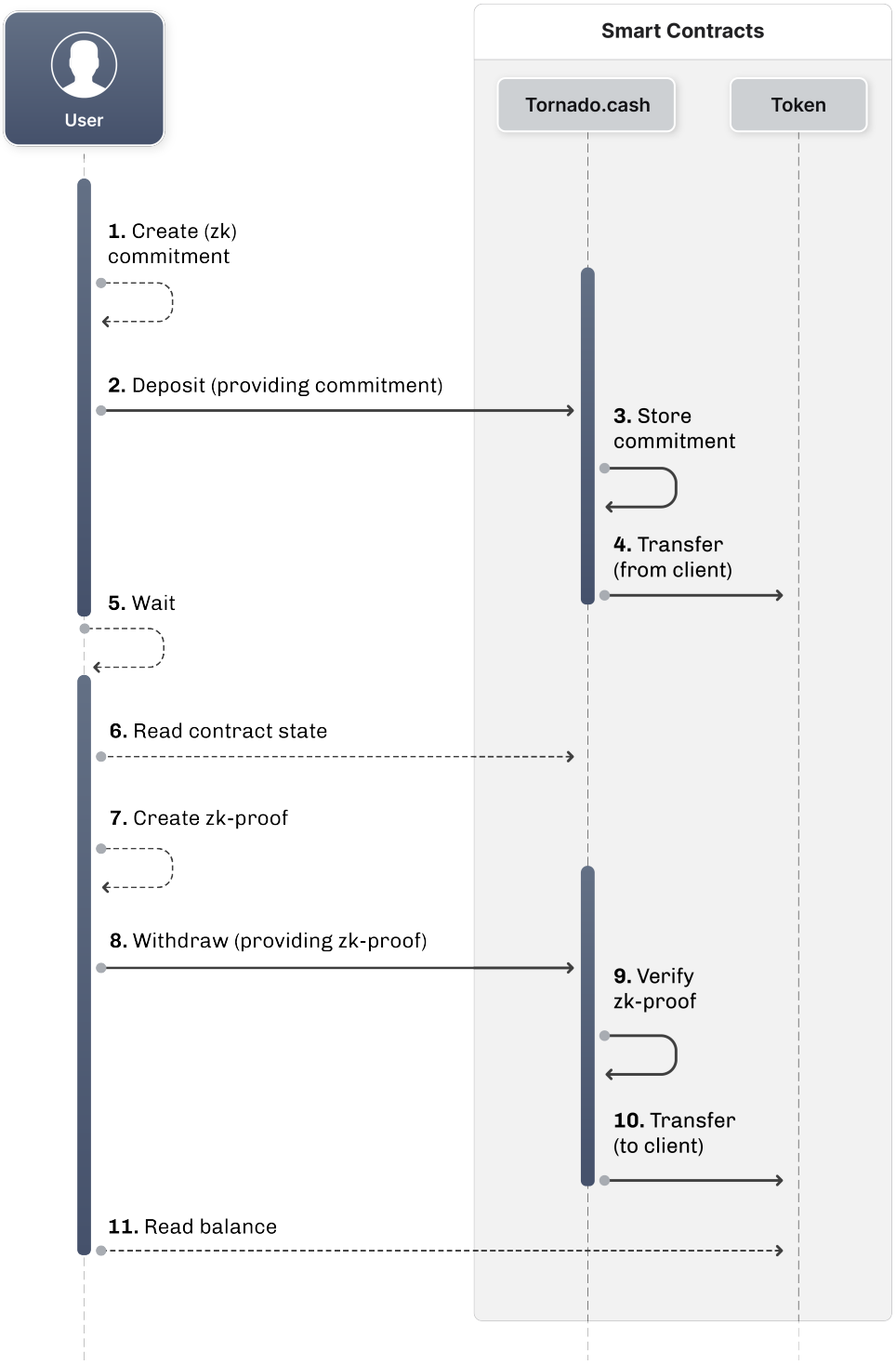

Figure 3: Example mixer solution implemented with smart contracts

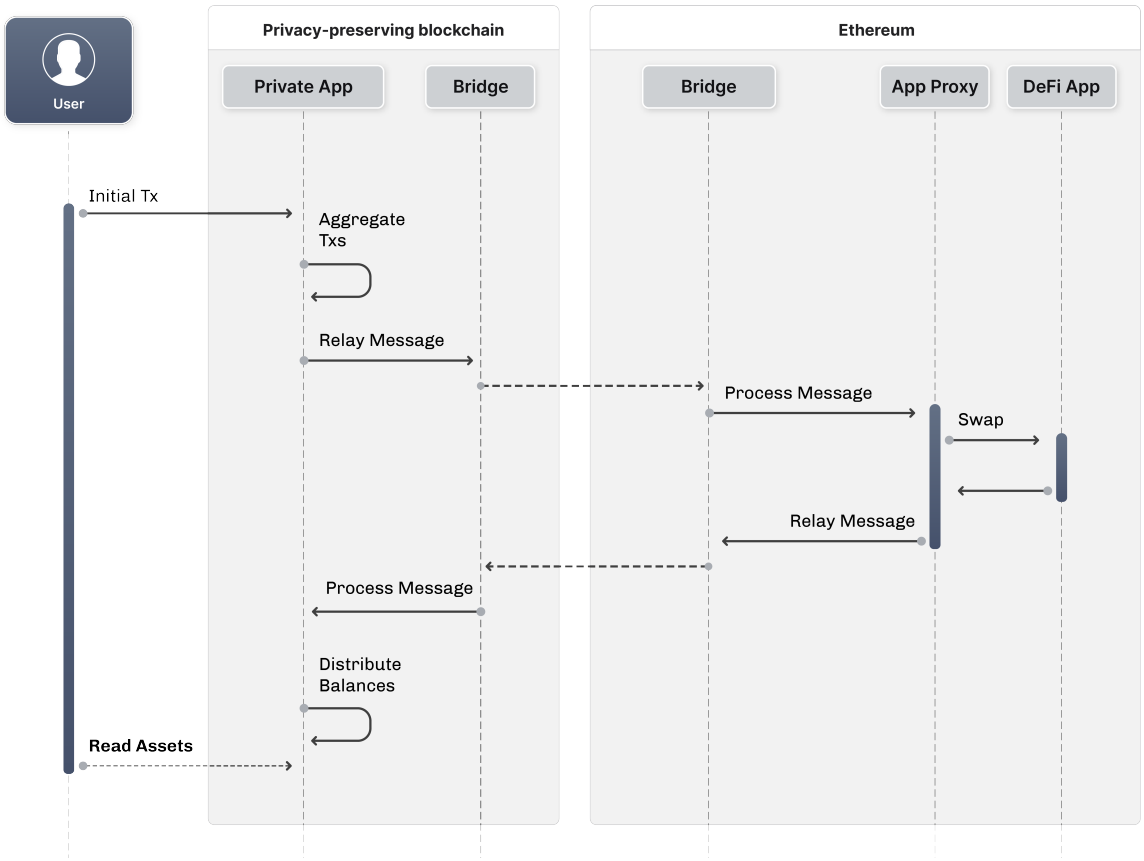

Figure 4: Example of a cross-chain architecture enabling privacy

6 Panther Protocol Architecture

6.1 Actors and Components

The protocol consists of the following actors and components:

- **Users:** Owner of cryptoassets who interacts with the Panther ecosystem via a wallet interface.
- **Peerchains:** Existing blockchains which Panther supports, and between which Panther can support private transactions.
- **zAssets:** Privately minted cryptoassets that are derivatives of other typically non-private cryptoassets (e.g. Ethereum ERC-20 tokens and similar). A zAsset is pegged 1:1 to the underlying cryptoasset which is locked as collateral in Panther Vaults; the latter may be partially or fully redeemed by burning some or all of the minted zAsset.
- **Panther Vaults:** Autonomous, zero knowledge, self-custodial smart contracts which act as decentralized custodians for collateral of zAssets.
- **Service Providers:** Entities that provide applications which support private transactions of zAssets by Panther users. Optionally, they may also request to verify attributes of

the users and their transactions, e.g. in order to satisfy regulatory compliance requirements.

- **Trust Providers:** Entities which provide verifiable statements *(attestations)* about users, which allow Service Providers to increase their level of trust in those users.
- **Panther Pools:** A collection of asset pools; each pool allows users to privately deposit, withdraw and transact on any peerchain through the use of one or more zAssets. All transactions are associated with a piece of specially formatted data which will allow users to voluntarily disclose details of transactions and link them (even retroactively) to attestations from Trust Providers. (See the section on Disclosure Mechanisms for more details.)
- **Panther Privacy Miners:** Participants in the Panther ecosystem which frequently contribute zAsset transactions to the anonymity set, and earn Panther Tokens as a reward.
- **Panther DAO:** A decentralized autonomous organization for protocol governance.
- **Relayers:** A network of proxy nodes for relaying transactions onto the peerchain in a privacy-preserving manner.

Figure 5 shows the high-level interactions between these different actors and components. For clarity, many details have been simplified or omitted.

6.2 Assumptions and Threat Model

Assumptions. For the user:

- User's device is secure and free of malware
- User takes precaution to protect their own network layer privacy, e.g. use Tor relay to randomize their IP address

Threat model. In our analysis we adopt a version of the *Global Passive Adversary* (GPA) threat model similar to that used in [20]. In summary, we assume a powerful adversary who

- is able to observe, record all traffic on all traffic between users, Panther mixer pools and third-party providers,
- is able to inject arbitrary traffic into any public networks and launch network attacks,
- is able to corrupt a minority of mix relays and observe all internal state therein
- is able to participate in Panther as a small number of users (below the Sybil attack [9] threshold)
- is computationally bounded, e.g. is not able to forge digital signatures
- does not perform attacks by exploiting security bugs in software

The adversary's goals may include:

- de-anonymizing transactions,

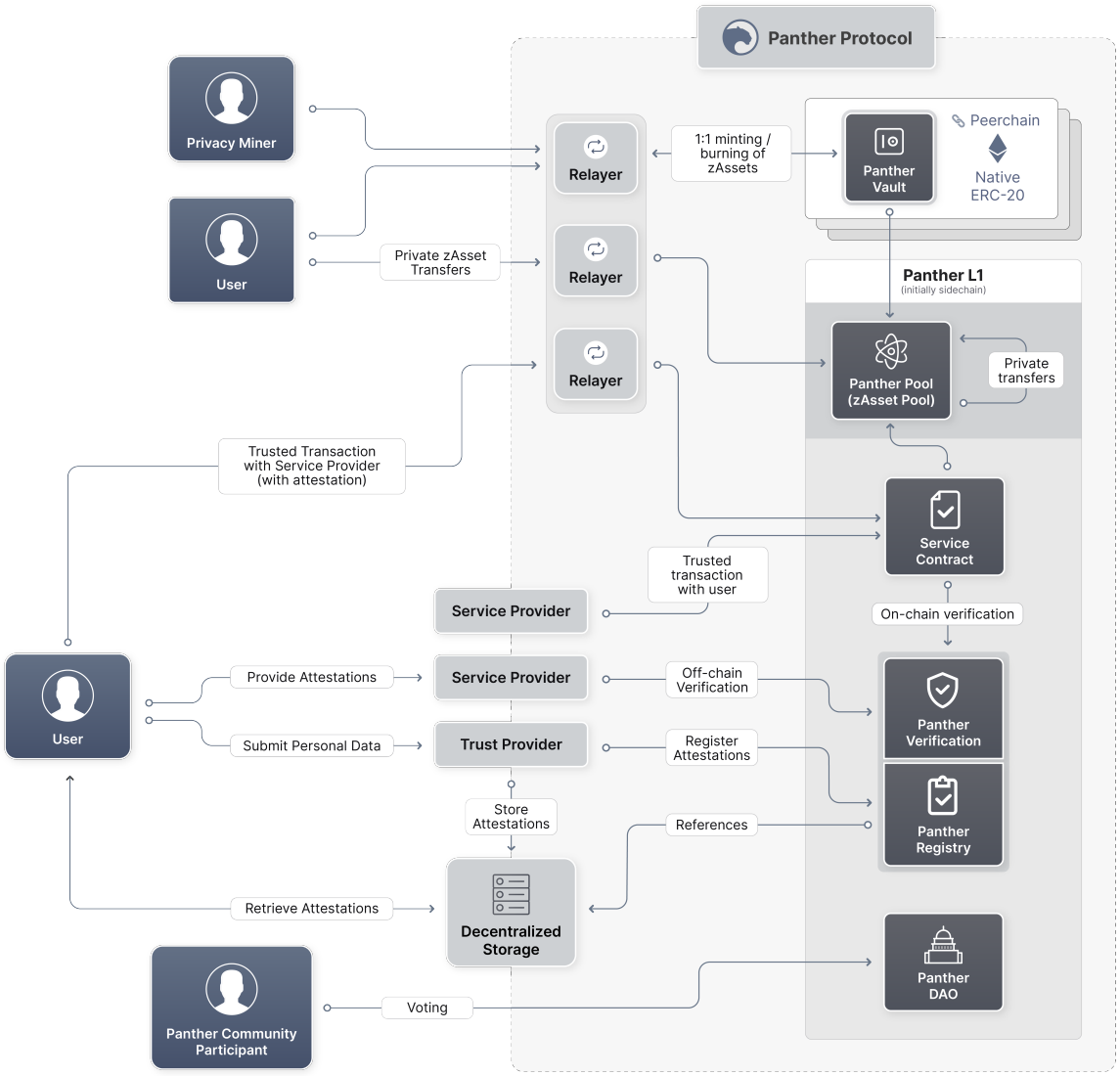

Figure 5: Panther Protocol Architecture

- minimizing as far as possible the anonymity set of a transaction,
- tracing back to the origin of a transaction,
- following the onward journey of a transaction,
- determining general transaction patterns e.g. volume, frequency, address reuse.

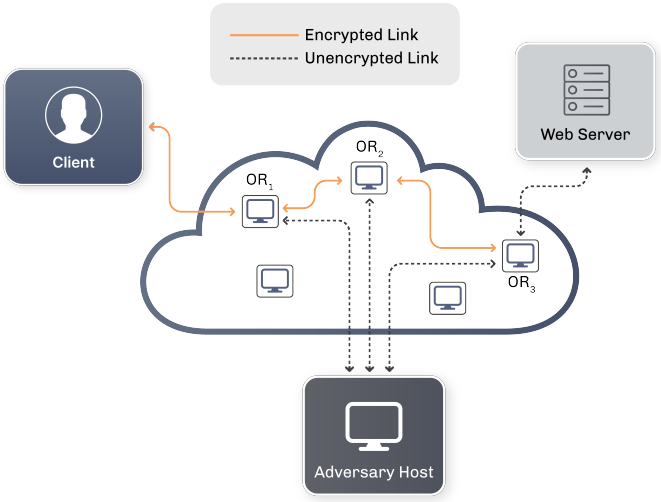

Figure 6: Global Passive Adversary Threat Model

6.3 Design Goals and Principles

Design Goals: Provide, within the Panther system of participants:

- Transaction unlinkability
- Transaction unobservability
- All participants are economically incentivized to behave honestly

Design Principles:

- Permissionless, allowing open participation
- Security and privacy by default
- Minimized trust assumptions
- Voluntary disclosures of private data
- Modular design by providing clean interfaces
- Easy integration with third-party services
- Upgradeable protocol without having to unshield already-minted zAssets

6.4 Enabling Trust whilst Preserving Privacy

Panther allows users to voluntarily generate and share disclosure statements with whichever third parties they choose, in order to prove:

- details of their interactions with Panther Protocol, and
- other statements about themselves which may be required in order to interact with counterparties.

The former is achieved by native functionality within Panther; the latter is achieved by Panther serving as a transport layer for data representing attestations about users originating from Trust Providers, together with tools for both off-chain and on-chain verification of those attestations, which can then be voluntarily disclosed by those users to Service Providers with which the users wish to interact. These voluntary disclosures establish a trust relationship of a Service Provider towards a user of their service, allowing them to interact privately whilst reducing the risk exposure of the Service Provider.

Whilst it is expected that the primary use cases motivating the reduction of this risk will be driven by the needs of Service Providers to comply with regulatory requirements, where it is assumed that the user already trusts the identity of the Service Provider, it is possible that the same mechanisms could also be used to establish trust in the opposite direction, i.e. from the user towards the Service Provider. Indeed, it is also conceivable that the model could be used to establish trust in user-to-user (a.k.a. P2P) transactions, or between Service Providers (a.k.a. B2B), or even in cases where private transactions are not required.

Trust Providers are typically publicly visible and reputable organizations. They could be banks, specialist KYC providers, certification authorities, government departments, notaries or a partner working on their behalf such as an electronic signature provider.

It should be noted that Panther operates on a permissionless model where anyone can become a Trust Provider, and it is up to the Service Providers to decide which Trust Provider(s) they will trust. If a Service Provider announces that they will accept equivalent attestations from multiple Trust Providers, then a user wishing to transact with that Service Provider also has some freedom in which of those Trust Providers to use.

Trust Providers are incentivized to be honest and provide true verifiable statements, also known as *attestations*, about users, by receiving payments from Service Providers or Users in Panther Tokens.[12]

Panther will make it easy for the users to receive and securely store these attestations in a decentralized manner, and to later retrieve those attestations and pass them to any Service Providers which need them. The attestations can be provided and verified either off-chain or on-chain; in the latter case, zero knowledge proofs are required in order to avoid public disclosure of confidential data.

Each Service Provider has different trust requirements, and therefore must be free to trust (or distrust) whichever Trust Providers they want. This implies that when receiving attestations from users, they must know which Trust Providers those attestations come from, so that they can make their own decisions whether to trust those attestations.

Figure 7 shows the interactions between the user, Trust Providers, Service Providers, and the Panther Protocol which makes it possible for the user to transact with Service Providers in a trusted context without loss of privacy.

Example use cases:

1. A Trust Provider performs KYC for a Panther user, and provides them with a digitally signed statements attesting any of the following:

[12] There needs to be a mechanism to disincentivize Trust Providers from making false attestations, such as some form of reputation tracking or other accountability. Initially this will be outside the scope of the protocol; however later Panther versions may introduce on-chain methods such as staking, DAO voting, slashing etc.

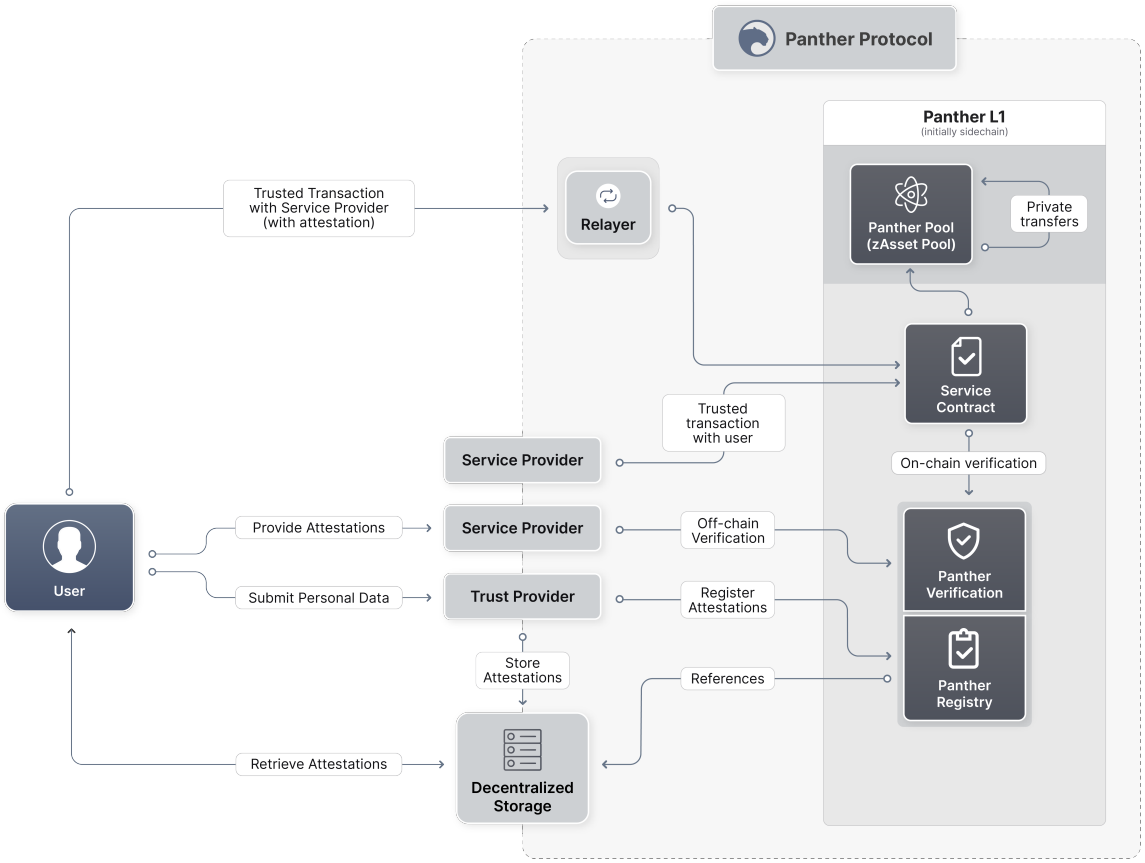

Figure 7: Trusted Transaction Architecture

- The user's passport photo is valid
- The user is a citizen of a given country
- The user has a given birthday
- The user is above a certain age

2. A Trust Provider verifies a driving license provided by a Panther user, and provides them with a digitally signed statements attesting:

   - The user is licensed to drive a minivan
   - The user has no unspent traffic violations

   These statements are received from the user by a car rental Service Provider which verifies the signatures to ensure the statements originate from the Trust Provider.

3. A manager of a security / token sale receives signed attestations showing that a Trust Provider certifies that an application to the sale is over 18 and a citizen of a selected list of approved countries, in order to accept zAssets as payment in return for tokens. They may also need to check that the participant is a qualified investor.

4. An under-collateralized DeFi lending protocol needs to check that a borrower has a minimum credit score before privately lending zAssets to them without requiring collateral to safeguard the loan.

6.5 User Journey

At a high level, a user might interact with Panther Protocol by taking steps such as the following:

1. Generate and register a cryptographically concealed anonymous user identity.
2. Deposit supported non-private cryptoassets and wait for them to be accepted into a Panther Vault, and the corresponding zAssets registered in the Panther Pool.
3. Once in the Panther Pool, the balance of the zAssets will be visible from the user's wallet.
4. The user may privately transfer any fraction of their balance to another user.
5. The user may withdraw any fraction of their balance into a new stealth address accessible by the user.
6. The user may deploy the new zAssets into DeFi protocols as they wish.
7. The user can disclose metadata of any of their transactions, e.g. to their accountant for tax purposes, to a centralized exchange or bank to satisfy compliance requirements, or to law enforcement.
8. Once exited a DeFi protocol, the user may:
    - deposit back into a Panther Pool to obscure its DeFi history,
    - redeem the zAsset for its underlying native non-private asset collateral by burning the zAsset. The withdrawn asset will be deposited into another newly generated stealth address.
9. Complete KYC and/or other checks with a Trust Provider, receiving signed attestations in return.
10. View those attestations from the user's wallet
11. Use those attestations and/or Panther transaction history to perform disclosures to counterparties.

7 Panther Protocol Implementation

This section presents some of the implementation plans and design decisions made at the time of writing. This whitepaper is a living document, and is liable to evolve as progress is made on the implementation.

7.1 Development Milestones

Our plan to build out Panther Protocol with a phased approach is summarized in Figure 8.

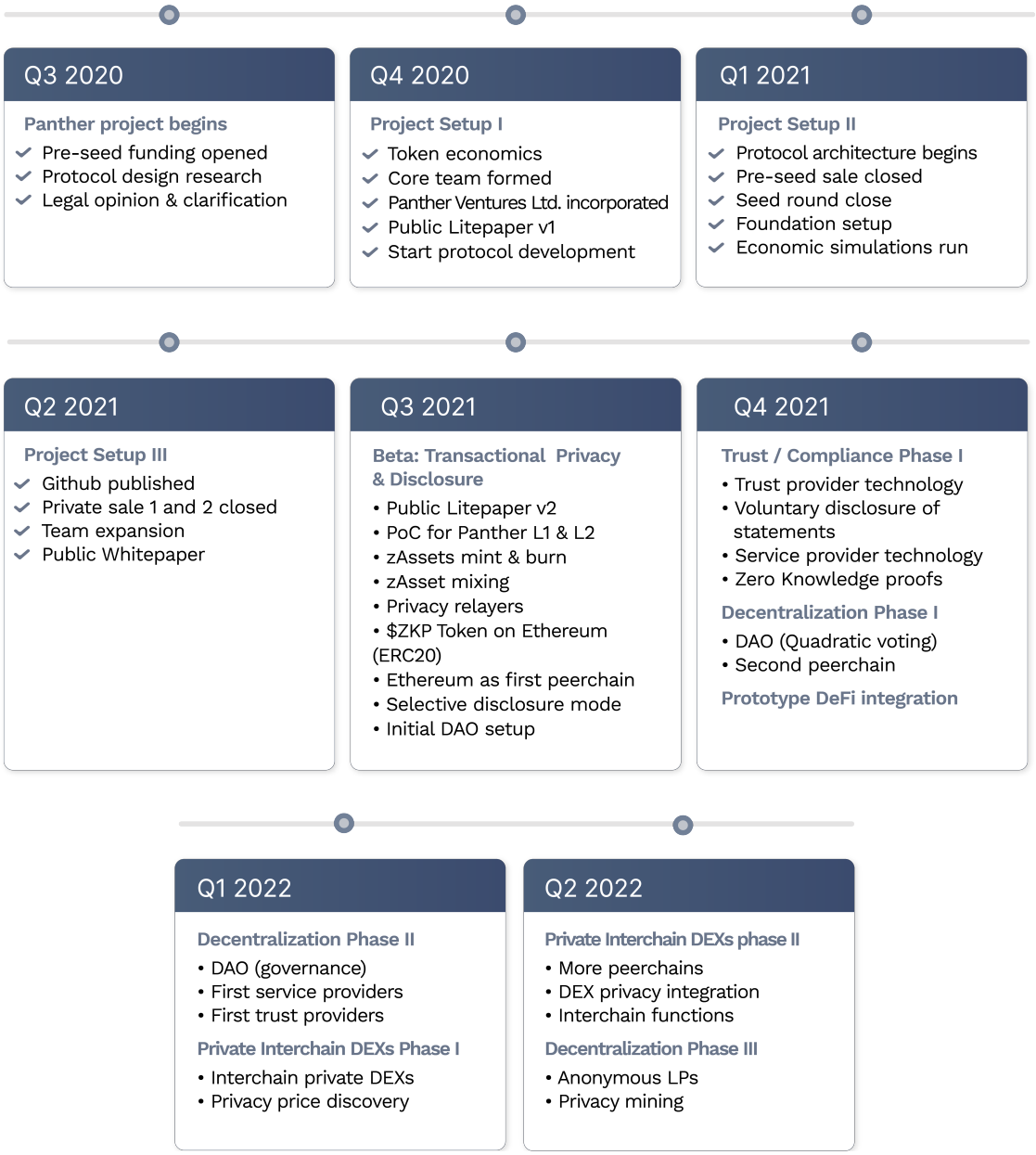

Figure 8: Panther Protocol Roadmap

In the Beta phase, we will introduce an initial implementation supporting core Panther functions of private zAsset minting, burning, and transfers, plus foundational capabilities around transaction disclosure.

The next phase will introduce basic DeFi interoperability, key features relating to trust and compliance, as well as community voting via the Panther DAO.

After this, new features for an on-chain governance mechanism will be added, together with the foundations of an inter-chain private Decentralized Exchange (DEX), the first Service Prov-

iders and Trust Providers.

Later phases will bring more advanced inter-chain functions, a privacy pricing mechanism, and eventually complete the Panther vision by adding support for anonymous liquidity pools and more peerchains.

7.2 Beta Functionality

The beta release will feature these design highlights:

- Panther facilitates privacy whilst retaining trust.
- Any user can privately mint, burn and transfer zAssets permissionlessly at any time, without being forced to prove anything.
- All zAsset transactions are associated with a cryptographically concealed, abstract representation of user identity.
- This concealed identity can be used by users to voluntarily and selectively disclose information about their prior interactions with Panther. It can also be used later (again, voluntarily and selectively) by users to retroactively associate those interactions with attestations provided by Trust Providers. This means that users can future-proof their private transactions by using Panther zAssets to proactively prepare for the possibility of incoming financial regulation and the corresponding compliance requirements.

7.3 Beta Protocol Sketch

From the initial beta release onwards, Panther Protocol will be a permissionless protocol which allows open participation in all activities.

Our current design uses mixers (smart contracts) on an Ethereum sidechain to implement Panther Pool functionality, with the Panther Vaults holding native ERC-20 assets on Ethereum Layer 1. We will provide a token bridge to allow transfer of tokens between the sidechain and Layer 1 DeFi smart contracts. In order to provide a seamless experience for the user, *stealth addresses and adapters* will be deployed to support operations for DeFi protocols such as Uniswap.

7.4 DeFi Interoperability

Figure 9 shows a sequence diagram illustrating a private swap transaction from zUSDT to zDAI with Uniswap.

In summary:

1. The amount of zUSDT to be swapped is withdrawn from the Panther zUSDT mixer on the sidechain
2. The zUSDT is bridged over to Layer 1 as USDT
3. The USDT is swapped for DAI on Uniswap

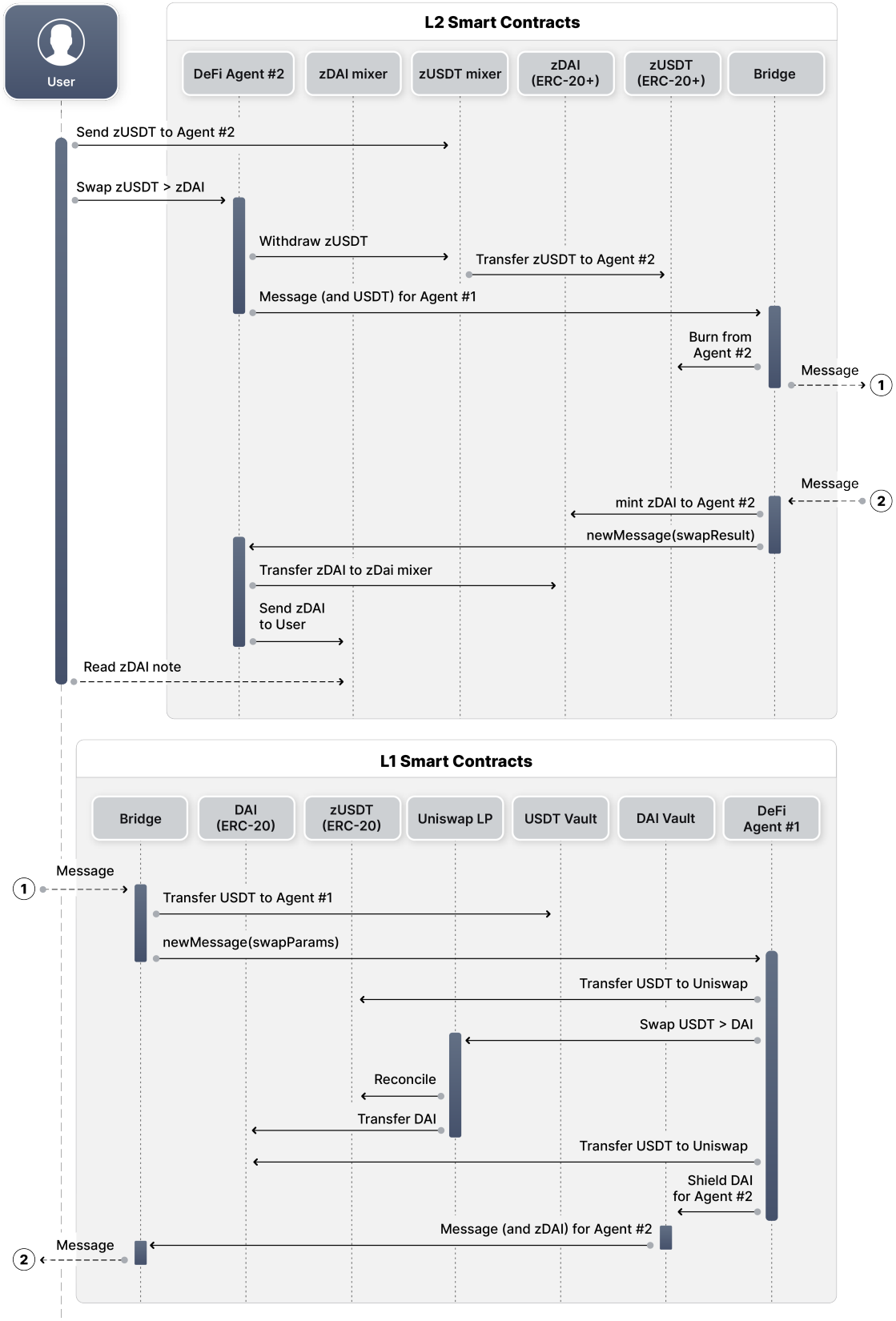

Figure 9: Panther Uniswap example - zUSDT to zDAI

4. The DAI is bridge back over to the sidechain as zDAI
5. The zDAI is deposited into the Panther zDAI mixer
6. The user obtains the deposit note for the zDAI

7.5 Disclosure Mechanisms

All disclosure mechanisms in Panther are private, secure, and voluntary. When disclosure of specific information is requested by a counterparty with which a user wishes to transact, the ramifications of the user choosing not to disclose are simply that the counterparty is likely to refuse further interactions with the user, although this is entirely at the discretion of the counterparty. This will not affect the user's rights to continue freely interacting with Panther Protocol and using their zAssets to transact with other counterparties.

There are several levels of disclosure which Panther will facilitate. Roughly speaking, these are likely to be implemented in increasing order of sophistication and levels of privacy attained:

- In early phases of the implementation, Panther will facilitate voluntary disclosure by users of the full details of selected subsets of their private transactions within the protocol. This mechanism will reveal that a subset of private transactions all involve the same user, and that the user owns given public accounts on the blockchain, without harming that user's privacy by linking other transactions to that user.
- As Trust Providers are introduced, the ability will be added to voluntarily disclose the full content of selected signed statements (attestations) from Trust Providers. For example, a user could choose to (privately) share with a counterparty an image of their driving license which has been notarized by a Trust Provider.
- Later, Panther versions will support more selective (fine-grained) disclosure which still discloses confidential unencrypted data to the requesting counterparty, but reduces the data shared to minimal levels. For example, if a user previously submitted their passport details to a Trust Provider, they could choose to disclose *only* a resulting notarized statement of their date of birth to a counterparty, without revealing their name or nationality.
- Finally, Panther will allow zero-knowledge proofs of signed statements from Trust Providers, and verification of such proofs (both off-chain and as a part of on-chain transactions), which will share no private information to the requesting counterparty or a smart contract beyond the absolute minimum which needs to be shared. For example, a user could prove that a Trust Provider has certified they are at least 18 years old, without disclosing their date of birth or any other personal information. We envisage that zero-knowledge proof would be most compelling for attestations of personal, private, or sensitive statements such as those about the user's age, wealth, or medical conditions.

In addition to the above, we will also explore the use of *interactive* versions of these proofs, which require the user to actively participate in proof verification protocols. Without the user's participation, a counterparty is not able to cryptographically convince a third-party that a previously verified proof is valid. This will allow the a greater level of privacy and *plausible deniability* which may be a desirable property.

8 Panther Cryptographic Protocols

Protocol design is currently underway. This section serves to indicate some of the areas under consideration; it will be updated periodically to reflect significant developments.

8.1 Stealth Address Protocol

We intend to use a Stealth Address generation scheme similar to Umbra [26], an explanation of which follows:

8.1.1 Setup

1. All users generate
   (a) Keypair for encryption/decryption: $ReadPubKey$, $ReadPrivKey$ keypair
   (b) Keypair for stealth address generation: $SpendPubKey$, $SpendPrivKey$
2. All users then register $ReadPubKey$, $SpendPubKey$ with Registry

8.1.2 Send Transaction

1. Sender retrieves recipient's $SpendPubKey$, $ReadPubKey$
2. Generate random R
3. Calculate $stealthAddress = R * SpendPubKey$
   (a) Note that the corresponding private key will be $R * SpendPrivKey$ which will only be known to the recipient
4. To ensure R is only available to recipient, we encrypt it with recipient's $ReadPubKey$ as follows:
   (a) Generate ephemeral (unique to this encryption) keypair $ePrivKey$, $ePubKey$
   (b) Compute Shared secret $SS = sha256(ePrivKey \times ReadPubKey)$
   (c) Encrypt by XOR to generate ciphertext $CT = R \oplus SS$
   (d) Note that $ePrivKey * ReadPubKey$ is basically Diffie-Hellman key exchange producing $ePrivKey * readPrivKey * G$, therefore only the recipient will be able to decrypt it to the correct R with $readPrivKey$
5. Send transaction to $stealthAddress$
6. Broadcast event $(stealthAddress, CT, ePubKey, token, amount)$

8.2 Zero Knowledge Proofs

ZKP will be used in Panther for

- proving correctness of minting/burning processes,
- proving the validity of statements (attestations) generated by a Trust Provider about a user, in order to help Service Providers establish trust in that user, and consequently fulfill their compliance requirements,
- other uses to be confirmed.

With recent rapid development in the field, there is a good selection of ZKP systems available on the market. The key selection criteria include the strength of security assumptions, size of proofs, proof generation and verification efficiency, gas cost (where relevant), code maturity and whether trusted setup is required. A comparison of three types of commonly used ZKP systems is shown in Table 2[13].

	SNARKs	**STARKs**	**Bulletproofs**
Algorithmic complexity of prover	$O(N * log(N))$	$O(N * poly\text{-}log(N))$	$O(N * log(N))$
Algorithmic complexity of verifier	$\approx O(1)$	$O(N * poly\text{-}log(N))$	$O(N)$
Communication complexity (proof size)	$\approx O(1)$	$O(N * poly\text{-}log(N))$	$O(log(N))$
Size estimate for 1 Tx	Tx: 200 bytes, Key: 50 MB	45 kB	1.5 kB
Size estimate for 10,000 Tx	Tx: 200 bytes, Key: 500 GB	135 kB	2.5 kB
Ethereum/EVM verification gas cost	$\approx$ 600k (Groth16)	$\approx$ 2.5M (estimate, no impl.)	n/a
Trusted setup required?	Yes	No	No
Crypto assumptions	ECDLP	PRNG	DL

Table 2: Zero Knowledge Proof Systems Compared

[13] Source: https://github.com/matter-labs/awesome-zero-knowledge-proofs

8.3 Inter-chain DEX Consensus Protocol

Panther supports inter-chain exchange of assets through the peerchain mechanism.

A peerchain consists of a number of nodes. Transactions are accepted when a peerchain's nodes reach a consensus. Table 3 provides a side-by-side comparison between the families of consensus protocols used by current blockchains.

	Bitcoin	Ethereum	Casper	Tendermint	Avalanche	Ouroboros
Family	Nakamoto	Nakamoto	Proof of Stake	Classical	Snow	Proof of Stake
Throughput[14]	$\approx$ 7 TPS *Protocol bound*	$\approx$ 14 TPS *Protocol bound*	$\approx$ 2,000-3,000 TPS	$\approx$ 1,000 TPS *Bandwidth bound*	$>$ 4,500 TPS *CPU bound*	$\approx$ 1,000 TPS per Hydra head
Finality	$\approx$ 60 mins / 6 conf.	$\approx$ 6 mins / 25 conf.	12 sec / 3 epochs	6-7 sec block time	$<$ 3 sec	20 sec
Energy Efficient	No *ASIC-Optimal*	No *ASIC-Optimal*	Yes *CPU-Optimal*	Yes *CPU-Optimal*	Yes *CPU-Optimal*	Yes *CPU-Optimal*
# of Validators	3 pools w/ $>$ 51% HR	2 pools w/ $>$ 51% HR	$<$ 4 million[15]	$<$ 200 w/o TPS loss	Thousands, no HR [16]	
Sybil Protection	Proof of Work	Proof of Work	Proof of Stake	Proof of Stake	Proof of Stake	Proof of Stake
Safety Threshold	51%	51%	51%	33%	80% parametrized	51%

Table 3: Consensus Protocol Comparison

9 Governance

9.1 Panther DAO

Panther's governance mechanism is aimed at decentralisation to the fullest degree over the course of evolution of the protocol. The aim is to implement mechanisms such as quadratic voting, so that community preferences are fully expressed. The roadmap (Figure 8) of the protocol includes staged deployment which allows transitioning of the governance from an initial centralized model to a fully decentralized one, a common approach in this industry. The *decentralized autonomous organization* (DAO) would ultimately have the ability to develop and accept improvement proposals, voting proposals shall implement mechanisms which reduces the changes of governance capture.

9.2 Treasury

In order to ensure that Panther protocol is sufficiently decentralized and its development is supported, we envision a treasury function to that end. We envision that once the DAO is bootstrapped and the initialization conditions are met, assets in treasury cannot be distributed or re-allocated without approval of on-chain governance. Developers can post the proposals in public. The voting of these proposals would happen, as per the governance mechanisms implemented.

[14] Best estimates from online sources and conversations with core developers
[15] Peak supply estimated at 120 million; minimum of 32 ETH per validator.
[16] Theoretically able to accommodate millions of participants.

9.3 Pricing and Equilibrium Analysis of Privacy

In this section, we will introduce a game notation, players, their incentives, describe the rules of interaction, and construct a dynamic model of a system. Analyzing system behavior for different sets of parameters helps us design optimal cost structures, in particular fees for using dark pools. Insights obtained from the analysis and numerical simulation of this system can be a tool used for the governance of the protocol.

Target function of the optimization problem addressed in this section is to minimize the probability of a system failure and maximize social welfare, in the lexicographic order.

Suppose there are n pools, $\{P_1, \cdots, P_n\}$. Each pool P is characterized with two attributes:

1. Cryptocurrency C_P from the set of supported currencies $\{C_1, \cdots, C_m\}$.
2. Size of the pool s_P.

Users arrive over the time and may choose one of the pools for their transaction. Arriving user A is characterized by 3 attributes:

1. Native crypto-currency it holds, C_A.
2. Size of a transaction he wants to make, t_A.
3. Intrinsic privacy parameter, p_A. This parameter measures how much is the user willing to pay for private transaction.

In the following, we will describe a list of assumptions to obtain a game dynamics.

- Any liquidity provider can leave, or join other pool, if he/she does not like the current pool. Joining other pool comes at a cost. If the new pool belongs to the same crypto-currency, then the cost is equal to the total costs of leaving and joining new pool. In case of joining other pool in the new currency, new cost, exchanging their currency in the outside market is added. We denote the combined cost of changing a pool as c_{P_o,P_n}, where P_o is a current pool and P_n is a new pool. That is, these costs have to be given as a matrix. While this matrix can be volatile, for the sake of simplicity we will assume that it is constant and take values at the beginning of the simulation.
- Users holding each crypto-currency arrive as in a Poisson random process. This process has convenient mathematical properties, which explains it being frequently defined in a time space and used as a mathematical model for random processes in different disciplines spanning from sciences like astronomy, biology, geology, seismology, physics to economics, image processing, and telecommunications. The intensity parameter λ_C is estimated from the data using statistical estimation methods.
- Liquidity providers arrive as in a Poisson random process. The intensity parameter λ_L is estimated from the data using statistical estimation methods.
- We assume that user sizes are distributed half-normally. Formally, if X follows an ordinary normal distribution, $N(0, \sigma_t^2)$, then $Y = |X|$ follows a half-normal distribution. That is, the half-normal distribution is a fold at the mean of an ordinary normal distribution with mean zero. σ_p is typically estimated from the data using standard statistical estimation tools.

- User privacy parameter p_A is distributed with a truncated half-normal distribution. Truncation happens at the point 1 and the parameter for the half-normal distribution is σ_p^2. σ_p can be estimated from the data using standard statistical estimation tools. Privacy parameter implies that the user values the privacy threshold t as $\frac{p_A}{t}$, that is, higher p_A means the user values privacy more.
- Both liquidity providers and users are rational and risk-neutral. This is a standard assumption in economics, to allow tractable analysis of the system. Behavioral elements in the players' decision making is left for future research.
- Constant marginal cost of processing a transaction of size t is a function of a pool size and transaction. It is (weakly-) increasing in the pool size and (weakly-) decreasing in the transfer. That is, the cost incurred by the user is equal to $c(s_P, t)t$. Formally, c is defined as $c : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ and $c(x,t) \geq c(y,t)$ if and only if $x \geq y$ for any t and $c(x,t_1) \leq c(x,t_2)$ if and only if $t_1 \geq t_2$. Determining a theoretically efficient function $c(\cdot,\cdot)$ is an ultimate goal of the Panther governance and is left to future research and experimentation.
- Each liquidity provider has own opportunity cost of locking up assets. We assume that for liquidity provider L, its constant marginal cost is c_L, which is distributed as truncated half-normal distribution.
- Privacy threshold is a decreasing function of a pool size. The simplest example of such function is a reciprocal, that is, the privacy threshold of a pool P with size P is equal to $\frac{p}{s_P}$, where p is some constant. General function is denoted as $p(s)$, where $p(x) \leq p(y)$ for any $x, y \in \mathbb{R}$ with $x \geq y$.

In the dynamics of the game we assume that users may choose different currency pools, if none of their native currency pools is suitable for them, or a pool of other currency improves his utility. Moreover, given a fixed crypto-currency of an arrived user, he/she may choose different pool size. Together with the assumption 9.3, these assumptions imply that the changes are done by the users in order to maximize their expected utility. Formally, arriving user A solves an optimization problem:

$$\arg\max_P U_A(P), \tag{1}$$

where $U_A(P)$ is an utility derived by the user A joining pool P. Note that here we assume a price discovery for each user implicitly. That is, each user determines which pool/threshold is the best for him and what price is he willing to pay for using this particular service.

Similarly, liquidity providers may choose either their native currency pool, or other currency pool, if they find their currency pools not optimal. Formally, arriving liquidity provider L solves an optimization problem:

$$\arg\max_P U_L(P), \tag{2}$$

where $U_L(P)$ is an expected utility derived by the liquidity provider L joining pool P.

Simulation is done in the following way. First, we generate entry points for the users of each currency according to a Poisson process. Similarly to users, we generate entry points for the

liquidity providers according to a corresponding Poisson process. Second, we sort these time points in an increasing way.

We go through all points in the increasing order. Once the user entry point is generated, we sample its currency. Third, for each user/point, we sample its transfer size and willingness to pay for the privacy. Depending on these parameters, we find optimal pool for the user, and if such exists – add this user's fee to the pool that is later distributed to liquidity providers depending on their sizes. Similarly for liquidity providers, we sample their sizes and opportunity costs to lock up their assets. If there is a pool which satisfies their requirements, they join to the optimal pool, otherwise, they stay out.

A new liquidity provider joining a pool has two effects on the current pool members. Firstly, it decreases portions of rewards existing members of the pool get. Secondly, it increases attractiveness of the pool to potential future users of the system for using this particular pool. Therefore, the total gains of the pool members increase. This trade-off is one of the important features of the system, and optimal choice of the cost function may decide if it is positive for the whole system or not in the long run.

In every time batch of fixed size, in our case, every month, we check how much each liquidity provider is earning. If gains in this time interval is less than standard opportunity costs the provider L pays for locking up the deposits, c_L, we assume that he will leave the pool. Liquidity provider leaving the pool makes the system worse. In this part of the simulation we assume that liquidity providers do gain discovery ex-post, once their gains are realized. However, some of them might not even join the system without knowing their expected gains ex-ante, given the costs of joining. In the future research we plan to design mechanisms for ex-ante gains' calculation.

We are interested in situations (sets of parameters) so that the system size (pool sizes) does not go to zero. The latter also depends on initial sizes of the pools. Given parameters of the system, there must be a vector of initial sizes of pools $s = (s_{P_1}, \cdots, s_{P_n})$, so that the system grows with positive probability.

Definition 1 *We call the set of points S in n dimensional Euclidean space $\mathbb{R}^n_+$ such that the system does not collapse to zero a* **lower bound set**.

Determining the lower bound set is a cornerstone of an optimal initialization of the pools. For generating LP's, we assign some number of LP's to each pool initially, with average parameters. Starting from the first time batch, these LP's act as other LP's joining to the pools, that is, they may also move to other pools or leave all pools altogether. We run the simulation for different initial vectors of sizes and observe the dynamics of the sizes. If the sizes do not converge to zero, we assume that they belong to a lower bound set.

The main parameter to optimize is the function $c(\cdot, \cdot)$, constant marginal cost of processing a transaction[17]. At this iteration we optimize it across the whole system. However, at later stages, the optimization can be done on the pool level, by the members of the pool.

[17] Code can be found here: https://github.com/pantherprotocol/simulations/blob/master/simulation.cpp

10 Tokenomics

The Panther Token ($ZKP) is a finite supply privacy-preserving governance token that represents a right to vote on governance proposals on the Panther Protocol. It is used in several instances to support the function of the protocol and provide incentives for its maintenance. The token has the following characteristics:

There will only ever be 1,000,000,000 $ZKP.

Later on in the process, $ZKP is used to reward Privacy Miners for providing zAssets to the Panther Pool, which could be viewed as a specialized form of liquidity mining. $ZKP is also used to pay relayer fees for new private Ethereum addresses.

In the early phases of Panther based upon Ethereum, $ZKP could be bought on Uniswap and used for the following:

1. To pay relayer node fees to responsible Gas Station nodes.
2. To compensate privacy miners, using open market pricing mechanics, for creating zAsset transactions within the Panther Pool on Ethereum.
3. To fund the Panther DAO and future Panther Improvement Proposals (PIPs) with a portion of transaction fees, at a rate set by $ZKP token holders.
4. To compensate Trust Providers for providing attestations about users.

Transactions and smart contracts are processed using native ETH to enable a simple user experience when using zAssets and interacting with DeFi.

$ZKP supply will be issued as follows[18]:

- 20 percent of total supply is issued to founders, team and advisors of Stellium, the company contracted to build Panther, with a gradual unlocking over 3 years.
- 20 percent of total supply will be sold by Stellium to the public with a gradual unlocking over 1 year.
- 10 percent of total supply will be sold by Stellium to private strategic investors with a gradual unlocking over 3 years.
- 40 percent of total supply will be reserved for staking rewards and emitted using an exponential decay issuance curve over 10 years to incentive early adopters and stakers on Panther Protocol.
- 10 percent of total supply will be reserved by the Panther DAO for community engagement and Privacy Mining incentives.

In the second phase of Panther, in the interchain DEX implementation, $ZKP could be used for paying various fees on the DEX.

In a later phase after the Panther DAO is created, governance decisions related to treasury management will move from Panther Foundation to Panther DAO.

[18] Correct at the time of writing but subject to change

11 Conclusions and Future Plans

In this paper we introduced the Panther Protocol, which serves to demonstrate that it is possible to combine privacy, trust, and composability in a single protocol, all of whose participants' correct behavior is informed and incentivized by well-founded game-theoretic models.

In the next phases, we will continue our journey to fulfil our vision. Panther Protocol will be developed to enhance the level of decentralization, and to become an industry-leading Layer 1 inter-chain DEX with privacy features.

This whitepaper is a living document and will be updated throughout Panther Protocol's development to reflect the current status of the project.

12 References

[1] Alessandro Acquisti, Curtis Taylor, and Liad Wagman. The Economics of Privacy. *Journal of Economic Literature*, 54:442–492, 2016.

[2] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Publishing, 3 edition, 2020.

[3] Guillermo Angeris, Alex Evans, and Tarun Chitra. A Note on Privacy in Constant Function Market Makers, 2021.

[4] Massimo Bartoletti, James Hsin yu Chiang, and Alberto Lluch-Lafuente. Maximizing extractable value from automated market makers. *CoRR*, abs/2106.01870, 2021.

[5] Jorge Bernal Bernabe, Jose Luis Canovas, Jose L. Hernandez-Ramos, Rafael Torres Moreno, and Antonio Skarmeta. Privacy-Preserving Solutions for Blockchain: Review and Challenges. *IEEE Access*, 7:164908–164940, 2019.

[6] Liad Blumrosen and Shahar Dobzinski. (Almost) Efficient Mechanisms for Bilateral Trading. *CoRR*, abs/1604.04876, 2016.

[7] David L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981.

[8] Whitfield Diffie and Susan Landau. *Privacy on the Line: The Politics of Wiretapping and Encryption*. The MIT Press, 03 2007.

[9] John R. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, pages 251–260, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[10] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[11] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A Pragmatic Introduction to Secure Multi-Party Computation. *Foundations and Trends® in Privacy and Security*, 2(2-3):70–246, 2018.

[12] FATF. Mitigating the Unintended Consequences of the FATF Standards. Available at https://www.fatf-gafi.org/publications/financialinclusionandnpoissues/documents/unintended-consequences-project.html.

[13] Alex Gluchowski. Optimistic vs. ZK Rollup: Deep Dive. Available at https://medium.com/matter-labs/optimistic-vs-zk-rollup-deep-dive-ea141e71e075.

[14] S Goldwasser, S Micali, and C Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery.

[15] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. SoK: Layer-Two Blockchain Protocols. Available at https://eprint.iacr.org/2019/360.pdf.

[16] Thomas Haines and Johannes Mueller. SoK: Techniques for Verifiable Mix Nets. Cryptology ePrint Archive, Report 2020/490, 2020. https://eprint.iacr.org/2020/490.

[17] Alastair Johnson. Is Privacy Under Threat From All The Know-Your-Customer Documents Stored With Countless Services? Available at https://www.forbes.com/sites/alastairjohnson/2019/04/03/is-privacy-under-threat-from-all-the-know-your-customer-documents-stored-with-countless-services/.

[18] Moxie Marlinspike. Technology preview: Private contact discovery for Signal. Available at https://signal.org/blog/private-contact-discovery/.

[19] A. Nitulescu. zk-snarks: A gentle introduction, 2020.

[20] Ania Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The Loopix Anonymity System, 2017.

[21] Ronald F. Pol. Anti-money laundering: The world's least effective policy experiment? Together, we can fix it. *Policy Design and Practice*, 3:73–94, 2020.

[22] Jonathan Fisher QC. Money laundering and financial exclusion. Available at https://brightlinelaw.co.uk/money-laundering-and-financial-exclusion/, 2018.

[23] Mikerah Quintyne-Collins. Awesome Privacy on Blockchains. Available at https://github.com/Mikerah/awesome-privacy-on-blockchains.

[24] Jan Christoph Schlegel and Akaki Mamageishvili. On-chain auctions with deposits. *CoRR*, abs/2103.16681, 2021.

[25] Bruce Schneier. Data is a toxic asset, so why not throw it out? Available at https://www.schneier.com/essays/archives/2016/03/data_is_a_toxic_asse.html, 2016.

[26] Matt Solomon and Ben DiFrancesco. Umbra: Privacy Preserving Stealth Payments On The Ethereum Blockchain. Available at https://github.com/ScopeLift/umbra-protocol.

[27] LexisNexis Risk Solutions. True Cost of AML Compliance Study. Available at https://risk.lexisnexis.com/insights-resources/research/2019-true-cost-of-aml-compliance-study-for-united-states-and-canada, 2019.

[28] Adam Spiers. Improving trust in the cloud with OpenStack and AMD SEV. Available at https://blog.adamspiers.org/2019/09/13/improving-trust-in-the-cloud-with-openstack-and-amd-sev/.

[29] Adam Spiers. OpenStack Docs: AMD SEV (Secure Encrypted Virtualization). Available at https://docs.openstack.org/nova/wallaby/admin/sev.html.

[30] Robert Stevens. DeFi: The Ultimate Beginner's Guide to Decentralized Finance. Available at https://decrypt.co/resources/defi-ultimate-beginners-guide-decentralized-finance.

[31] The Harris Poll. IBM Survey Reveals Consumers Want Businesses to Do More to Actively Protect Their Data. Available at https://theharrispoll.com/ibm-survey-reveals-consumers-want-businesses-to-do-more-to-actively-protect-their-data/.

[32] Alexander Viand, Patrick Jattke, and Anwar Hithnawi. SoK: Fully Homomorphic Encryption Compilers, 2021.

[33] A. C. Yao. Protocols for secure computations. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 160–164, Los Alamitos, CA, USA, Nov 1982. IEEE Computer Society.

[34] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. Profile Books, 2019.