THE IDEA OF PIXELCASH

BY KYLE D. USHER

INTRODUCTION

Pixelcash is a conceptual PoW (proof of work), truly decentralized, peer to peer, and completely anonymous cryptocurrency with smart contract (built-in escrow) capabilities. It can also be called "Pixelnote" in regards to the units of value themselves. See the Pixelcash logo here: https://drive.google.com/file/d/1RWYjKPWEPqHOltP_kppFOIra7_WXtSXn/view?usp=drivesdk. The keyboard symbol for Pixelcash could look something like this: 【P̃】. Hopefully, though, Pixelcash doesn't turn out to be yet another "shitcoin", as there are already too many of those on the market being perpetrated by the likes of these two fine fellas: https://youtu.be/Z4gZGHP1y8U.

Let me just be honest and say that Pixelcash (PXC) was conceptualized as another one of those "novelty" cryptocurrencies, like Dogecoin (DOGE). If you think about it, cryptocurrencies are a lot like fireworks, in the fact that there are thousands of varieties of both, and many of the varieties are themed, and many varieties are simply for fun. Pixelcash would be one of those varieties. Pixelcash basically takes the old "needs more jpeg" meme from the early 2010s (see here: http://needsmorejpeg.com) and turns it into something more modern and useful. They say the a picture is worth a thousand words. Well, because of Pixelcash, now it's worth money. The official tagline for Pixelcash is "Pupa Pictura is Minarum Mille", which is a Latin phrase that literally translates into "A Picture is Worth a Thousand Dollars".

MINING

A whole Pixelcash unit (one Pixelnote) itself starts life as a grouping of exactly 1,000,000 (one million) freshly minted, high quality .jpeg format images, hosted in a cloud, or more specifically, a Distributed Data Center, or DDC, which will be addressed further in this paper. These images are proprietary renderings of a superficial “coin" featuring Pixelcash branding referred to as a “jpenny" (further explanation on this term below) that are directly embedded into the protocol and copied by the miners as the method of creating new units. There are two different variations of the coin design, which can be viewed here:

1. https://drive.google.com/file/d/1sxq5VWn19iK248akdzlNBTbTHGNy0kVd/view?usp=drivesdk
2. https://drive.google.com/file/d/1x8F_pyI9aLVQYGiqfpCcLCuU_UpESf0Z/view?usp=drivesdk

The task of running software that copies and encrypts these images into the cloud is a component of the “mining” scheme, and just like (early) Bitcoin mining, anybody with a laptop or gaming PC can join in. However, unlike Bitcoin, the Pixelcash mining algorithm will be programmed to be ASIC resistant, i.e. remaining resistant to ASICs through regularly changing its adaptive mining algorithm, as ASICs cannot adapt to change. Choosing the right consensus tool is key to keeping Pixelcash decentralized and in the hands of the entire community, not just people with the most powerful machines, ASICs cannot achieve this. This is one of Bitcoin’s main disadvantages. Pixelcash mining would also probably be a more suitable alternative (as opposed to Bitcoin or Ethereum mining) for the many homebuilt mining rigs that utilize GPU graphics cards, such as Nvidia, not only because Pixelcash mining is ASIC resistant, but also for the fact that it involves actual graphics, albeit in a different format. It is not clear what the unit of mining power would be called in this case. For example, Bitcoin typically uses “terahashes per second”, or TH/s, as a measurement of power, because the SHA-256 algorithm is, well, a hash. Other cryptocurrencies, like Bitcoin Gold, use “solutions per

second”, or “sols/s”. Perhaps because .jpeg uses a lossy form of compression based on a mathematical operation called the “discrete cosine transform”, or DCT (see this article: https://en.wikipedia.org/wiki/Discrete_cosine_transform), the unit of mining power for Pixelcash could be called “transformations per second”, expressed as “trans/s”.

Each .jpeg image is assigned a unique serial number or other identifier by the protocol, and the exact arrangement of pixels within the image, the date and time it was mined, and it's date and time of mining are also recorded. Pixelcash has six decimal places, displayed as 0.000000. The smallest unit of Pixelcash is called a “jpenny", which is a portmanteau of the words “jpeg” (image format) and “penny” (smallest unit of the U.S. Dollar. It is one one-millionth of a Pixelnote.

SPENDING

When Pixelcash is spent from a wallet, the individually encrypted images (jpennies) are decrypted, and each is matched against the data saved earlier in the cloud’s ledger (i.e. the serial number and exact arrangement of pixels within the image), and once everything checks out, the jpennies are processed by the cloud's algorithm (more detail on how below) and delivered to the recipient’s wallet, before being encrypted again. Just like Bitcoin, odd amounts of Pixelcash can be spent, for example; if 25.327496 (twenty five point three-two-seven-four-nine-six) Pixelnotes were spent, that means that 25,327,496 (twenty five million, three hundred and twenty seven thousand, four hundred and ninety six) individual jpennies, each with their own unique serial numbers, would be processed by the cloud’s ledger and “given more jpeg”, or compressed into a reduced file size, incurring loss of information and introducing compression artefacts (lower image quality) as a result of the “lossy compression” characteristic of the .jpeg format. There would also be a small transaction fee awarded to the miners.

After each transaction, the exact pixel arrangement of each and every jpenny is again recorded, this data is saved again to the cloud ledger, and the cycle repeats when the Pixelcash is spent again. Compressing these large amounts of .jpeg images and processing their credentials to verify and complete transactions are the other component of Pixelcash mining, in addition to copying the original images to be used as jpennies from the protocol.

Unlike Bitcoin, Pixelcash transactions are completely anonymous, as the public Pixelcash ledger is private. The details displayed on the public ledger are the transaction ID (copies of which are included in both of the users' private transaction histories for their own reference), the amount of Pixelcash sent or received, the date and time of the transaction, the coinwall (more details below) and the size of the transaction in either megabytes, gigabytes, terabytes, or petabytes. For example, the file size of one of the jpenny designs is roughly 588 kB, or kilobytes. If a whole Pixelnote is comprised of 1,000,000 such images, that means a transaction of one whole Pixelcash unit would "weigh" around 588 gigabytes, or 588 GB, in size. Of course, because fractions of Pixelcash can be spent, the ledger's algorithm would need to choose the appropriate prefix (mega, giga, tera, or peta) in regards to the number of bytes in each transaction. Megabytes are the smallest unit that can be used for this, for the simple fact that not less than 100 jpennies, with a combined file size of roughly 58 MB, or 0.000100 Pixelcash, could be spent at a time due to network (miner) fees.

Naturally, because the file size of each jpenny is reduced with each transaction, the transaction size for those jpennies will be displayed on the ledger as being smaller and smaller until the end of their life. For example; since one newly mined Pixelnote has a combined file size of roughly 588 GB, a transaction of the same amount of Pixelcash that only has a transaction size of 3.2 MB could indicate that the jpennies that make up the unit are nearing the end of their life. This is an important detail to include in the ledger, especially for recipients, to help them gauge the viability of their funds and request new payment in fresh Pixelcash from the sender if desired. Wallet addresses or their final balances are not displayed, similar to the Monero ledger. The coinwall is a proprietary feature of the public ledger which adds an extra layer of transparency to the transaction. It is

comprised of a snapshot of all the jpennies in a transaction hosted on a single webpage that can be viewed with the naked eye, as an excerpt from the Distributed Data Center, which displays the different states of jpeg compression (from low data loss/high quality to high data loss/low quality) that each jpenny is currently in at the time of the transaction, providing an enhanced sense of fungibility and serving as proof that the currency operates on the premise described in this paper. This means that if 0.010000 Pixelcash are dealt in a transaction, there would be 10,000 images available for human viewing under that transaction on the ledger. While it is unlikely that many people, if any, would actually view all of these images, it does, as already mentioned, offer further fungibility, transparency, and verification, similar to the "show scripts and coinbase" feature present on some Bitcoin block explorers such as blockchain.com.

Due to the nature of this particular system, it would not be computationally practical for a full client wallet to exist. Full client wallets verify transactions directly on a local copy of the blockchain saved on the user's device. The file size of 1,000,000 Pixelnotes alone would weigh over 588 petabytes, or 588 PB, which is far more than any household PC and even the entire Sia cloud (https://sia.tech) can store. Therefore, only lightweight client wallets (such as smartphone apps like Coinomi), online wallets, paper/physical wallets, and hardware wallets would be practical for holding and spending Pixelcash. An example of a Pixelcash paper wallet can be seen here: https://drive.google.com/file/d/101zltxfIHLXRX9S8FlMRFdznQYIIZTRn/view?usp=drivesdk. Pixelcash smart contracts would use a form of escrow dubbed "Mutually Assured Destruction" (MAD) escrow (see here: https://bitcoinmagazine.com/articles/particl-takes-mad-approach-escrows-maximizing-privacy/).

If you're not familiar, a smart contract is an agreement that can be enforced through a blockchain. Rather than relying on trust or a legal framework to ensure that each party that enters into a contract will adhere to its terms, you can use the blockchain to create a contract that is automatically enforced, between two people, in a decentralized fashion. Ethereum has become the most popular blockchain for creating smart contracts. One of the major design goals of the Ethereum platform

was to support smart contracts. From the start, this set Ethereum apart from Bitcoin, which was created first and foremost as a digital currency platform. Both lightweight client and hardware Pixelcash wallets would preferably have the smart contract capability built in. These wallets would enable smart contracts basically by locking (temporarily rendering unspendable) deposited funds until all of the parties sign off on the transaction. MAD escrow is a technique that effectively prevents fraud in a transaction without requiring the oversight of a third party. In a MAD escrow contract, a buyer and seller both place funds into escrow. The seller starts by depositing an amount they want the buyer to match to symbolize a virtual handshake. This could be between 1 and 100 percent of the item's purchase price. The seller then creates an "invoice" for the transaction, which is essentially the contract itself, containing the seller's payment address and the exact amount required to complete the transaction. A copy of the invoice is then sent to the buyer either in the form of an auto-generated alphanumeric code (ex. A1BB23C4D567E890) which is the transaction ID itself, and is to be copy and pasted into the "pay to" field of the buyer's wallet, or as a QR code that is to be scanned. The buyer then deposits an amount equal to the handshake amount plus the price of the item they are buying. The escrowed funds are not released to anyone until both parties confirm that the transaction has been completed satisfactorily. When both parties do confirm this, the smart contract will be fulfilled and will be displayed as "confirmed" on the ledger. If a dispute arises, and the transaction is ultimately canceled, then it will be displayed as "unconfirmed". In either situation, the status of the contract is final and cannot be altered or supplemented in any way, ex. if a seller wanted to ask for more money later on, the original invoice and corresponding transaction ID would at that point be rendered invalid and could not be used to receive any further payments; the seller would have to create a new contract. This technique prevents either party from profiting through cheating in a transaction. With this approach, buyers and sellers using the Pixelcash network can operate without worrying about fraud or paying unnecessary fees. They also don't have to sacrifice privacy because no third party is involved in the transaction. Furthermore, and perhaps most significantly, because there is only basic scripting involved, security concerns are minimal. While Ethereum provides more extensible support for smart contracts, that flexibility comes with a higher risk of security and privacy threats. The more code that goes into a smart contract, the greater the risk of introducing a vulnerability that could enable an intrusion. Pixelcash would serve as the foundation for a completely decentralized platform

that supports a multitude of decentralized applications (dApps) and programmable functionality while offering high anonymity at the same time.

THE NETWORK

It is currently unclear as to exactly how the Pixelcash cloud itself would be maintained, i.e. who or what would provide the massive amount of storage space needed. With proper funding, large data centers like those owned by Google could be built and completely dedicated to the Pixelcash project, or storage space could be purchased from many different providers. An unconventional, yet perhaps more immediately feasible approach would be to put members of the Pixelcash community itself in charge of maintaining the cloud, by enabling them to operate through a concept called the "Decentralized Data Center", or DDC.

The DDC would be comprised of a large network of homebuilt miniature data centers, or "nodes" that are all synced with the Pixelcash network and store a certain number of jpennies based on their storage capacity. These nodes would consist of multiple external hard drives (such as this model: https://www.wdc.com/products/external-storage/my-book-duo.html) all connected together via a USB hub (such as this model: https://www.sabrent.com/product/HB-U14P/13-port-usb-2-0-hub-power-adapter/) and plugged into the node operator's main USB port on their PC. In this particular example, the 13 port USB hub could host up to 13 of the 20 terabyte (TB) hard drives, making for a total storage capacity of 260 TB, enough to store roughly 442 whole Pixelnotes.

Ideally, there would be hundreds of thousands of these homebuilt nodes all around the world, operated by supporters of the Pixelcash project (which could be anyone), and there could be very many variations of them; from even larger versions of the example shown here, to a single common flash drive plugged into a laptop. The PC would have a special program created by the Pixelcash developers installed on it that detects the node and syncs its available storage space with the

Pixelcash network, allowing the network to interact with it as needed, i.e. by encrypting the hard drives for security (if they are not already encrypted), saving the .jpeg files to the hard drives and recursively compressing/deleting them as needed, etc. Other than keeping the PC and the node powered on 24/7 (each external hard drive would need its own power supply), no maintenance is required from the node operator.

Most likely the easiest way to achieve this in the early stages of the project would be to create a web application that allows the Pixelcash network to sync with any online cloud storage account, i.e. Google Drive, Microsoft OneDrive, DropBox, etc. and interact with it in the same manner as a physical external hard drive would in order to support the network. The specific cloud storage account(s) in question would be required to be completely clean before it could be accepted by the network, i.e. would need to be a new account(s) created specifically for this purpose, and not contain any personal items. Anyone could download/use this application, sync it with any (preferably free) cloud storage service, and that's it. It would be completely "set it and forget it". This application may also be simply run by a script that automatically registers new cloud storage accounts based on network need with cryptographically generated usernames and passwords for the purpose of preventing human access to the accounts. A script to do this task would be much more efficient than relying on human creation of accounts, as the network would be able to "help itself" to whatever amount of storage space it needs as mining takes place. For example, if one Pixelnote is roughly 588 GB in size, and a free Google Drive account offers 15 GB of storage space, the script would have to open 40 new accounts for each Pixelnote mined, using two different randomly generated strings of characters as usernames and passwords, such as "BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2" (username) and "c1qar0srrr7xfkvy5l643lydnw9re59gtzzwf5mdq" (password). Although it may be more economical for the developers, this approach has the obvious drawback of adding centralization by relying on multiple centralized entities becoming part of the cloud to keep the network up and running, versus simply storing all data within professionally owned and operated commercial data centers or dedicated physical storage. If one of the homebuilt nodes were to be destroyed for any reason, then all Pixelcash units with data stored in that node would be lost forever, being rendered unspendable as a result of permanently losing contact with the network. In this

respect, this is the only way that Pixelcash could be "burned" like Bitcoins. The ledger would be required to report these missing units on the coinwall, perhaps by displaying a simple red "X" icon in place of all missing jpennies.

Pixelcash wallet addresses and private keys would be long strings of randomly generated alphanumeric characters, similar to the ones automatically generated as file names for folders downloaded from external sources as a result of the MD5/SHA-1 functions of other systems (see here; http://www.tomshardware.com/answers/id-1752853/folders-long-random-words-letters.html). An example wallet address could be 1cc257829bebe0b3188a62beb7. Unlike many other cryptocurrencies, Pixelcash wallet addresses and private keys would be devoid of any universally known prefix, such as the "1" at the beginning of every Bitcoin address, which further strengthens Pixelcash's anonymity by providing ambiguous addresses. Each Pixelcash address can begin with a different number or letter. This is simply to fit in with Pixelcash's overall theme. Since Pixelcash is cloud based, wallet addresses and private keys act just like normal email addresses and their respective passwords would, i.e. they are credentials generated as a pair by the platform and used as allocation and authentication tools. Because of this, it would be impossible to "burn" Pixelcash in the traditional sense, i.e. destroying it by sending it to a fictitious wallet address, because the units are hosted within a cloud, and just like email messages, they cannot reach the recipient if the wallet address is invalid, and the client software will also not allow sending without a valid wallet address.

SUPPLY

There is no hard cap on the total number of Pixelnotes that can be in circulation. Miners simply earn an amount directly proportionate to their computing power. However, due to the fact that after being spent a certain number of times (be it hundreds or even thousands, see this video: https://youtu.be/NzsbjwuWYYI), the amount of data within a single jpenny would be reduced so far that the image cannot be compressed any further; it would be a single, monotone colored block completely unrecognizable on the coinwall, and thus it would be rendered unspendable by the cloud's algorithm and automatically deleted before it could be spent again, enforcing a notion of scarcity. A Pixelcash wallet client would detail exactly how viable a user's funds are, i.e how many times the units can be spent, the exact percentages of each jpenny in the wallet at a different stage of its lifespan (rated on condition; Mint, Good, Fair, or Poor), and would automatically allow the user the option to spend the best percentage of their wallet balance. This is synonymous with paper money becoming so worn out and unrecognizable over time that it is deemed unacceptable by merchants, banks, vending machines, etc. and must be discarded. Perhaps this limited lifespan could serve as a motive to HODL Pixelcash. Conversely, the prospect of a user's Pixelnotes being burned as a result of a hard drive being damaged or cloud account being closed could serve as a motive to spend it as quickly as possible, just like real cash typically is. In preparation for burns, however, it would perhaps be intuitive to program the network to execute automatic refunds to wallets that are affected by these circumstances, synonymous with the U.S. Bureau of Engraving and Printing offering citizens new banknotes in exchange for mutilated ones. To prevent fraudulent refunds (i.e. users intentionally disconnecting hard drives or cloud accounts that they know contain their Pixelcash, getting a refund, then reconnecting them again to end with twice the original amount in their wallet), the network must recognize that a refund has already been given, and hold the previously lost Pixelcash in a master escrow until it is needed for distribution to other users of the network that are due refunds. Due to the fact that the demand for refunds could possibly outweigh the supply available from this method alone, the master escrow will be funded with 10% of each miner's profit, synonymous with taxes being withheld from regular income by federal or state governments. This 10% would be automatically deducted by the network before the proceeds are deposited into the miner's wallet, meaning for every 10 jpennies that are mined, 1

is taken by the network and the remaining 9 are to be kept as profit by the miner, and jpennies are deposited into wallets in intervals of 9. Any wallets due a refund would be immediately credited with these deductions. In the case of many refunds being due concurrently, the total amount available in the master escrow will be divided up evenly among all of the wallets owed until each is gradually paid off. Each jpenny held in the network escrow would be encoded into a Base64 string and saved as a .txt (text) file within a separate portion of the DDC (Distributed Data Center) cloud or a sidechain. An example of Base64 encoding can be found here: https://www.base64encode.net. Due to the extremely small file size of .txt files (converting an image of kilobytes into mere bytes), a copy of the entire network escrow could potentially be saved on a single hard drive. With this in mind, copies of the network escrow sidechain may be saved in multiple places within the DDC, making a loss of the network escrow virtually impossible. When the escrow is needed, the Base64 strings would be decoded back into .jpeg images and released into the mainchain. Likewise, jpennies recovered from a temporary network absence would be Base64 encoded and moved to the sidechain.

According to Forbes (see this infographic: https://www.forbes.com/sites/niallmccarthy/2014/09/12/how-many-years-do-us-banknotes-stay-in-circulation-infographic/), the United States banknote with the longest average lifespan is the $100 bill, at 15 years, due to the simple fact that they are often retained for their value and only used for larger purchases. Other banknotes, like the $5 bill, last only 4.9 years. Based on this data, it could be assumed that jpennies being consecutively used for smaller transactions (such as buying coffee, lottery tickets, small purchases online, etc.) would “wear out” sooner than whole or large portions of Pixelcash units used for more expensive purchases. It is hard to tell, however, just how long in years that individual jpennies will last being transferred from peer to peer. As the video demonstrating recursive .jpeg compression clearly showed, a .jpeg image on the highest quality settings can be compressed over 2,000 times and even still be visually recognizable. Even the unrecognizable images could be further compressed, until all pixels within the image eventually assumed the same color and grouped into a large singular block. A Federal Reserve survey found that physical currency turns over about 110 times a year, i.e. 2 times a week. However, it is possible that cryptocurrencies are turned over much more often than that. Because the jpenny

designs do not have as near as many graphic details as the paintings used in the compression demonstration, they would likely not last through as many compressions.

Pixelcash would be the only cryptocurrency to directly emulate physical fiat currency while being decentralized at the same time, save for the fact that Pixelcash, just like any other cryptocurrency, cannot be counterfeited. The total supply is also capable of keeping itself proportionate, for example; if the number of jpennies minted each year increases as a natural consequence of a growing adoption of Pixelcash, but the number of people spending Pixelcash also increases for the same reason, then there will ultimately be an equal number of jpennies eventually being destroyed as there are being mined.

MORALE

The cryptocurrency community seems to believe that in order for any newly created cryptocurrency to gain traction and become a success, it must not only be unique, but solve an existing problem. One notable problem in the current cryptocurrency community is circulating supply (aka scalability) and market cap. Some cryptocurrencies like Bitcoin have a limited circulating supply, while others like Ethereum have an unlimited circulating supply. Regardless of their circulating supply, however, all cryptocurrencies currently on the market have created monetary value from nothing, quite literally “just numbers on a screen”. Once again, Pixelcash is the only cryptocurrency that emulates physical currencies in the fact that its network limits the total circulating supply by enforcing a limited lifespan that directly correlates to how many times they are transacted, giving the currency a much more “real” feel. In addition, due to their open source nature, most other cryptocurrencies can be either cloned or hard forked. Hard forks tend to devalue the original currencies they were forked from, because Pixelcash, while also being open source, is not entirely clone or hard fork proof, but it is highly resistant, for reasons clearly explained in a second paper on this topic authored by me (see here:

https://drive.google.com/open?id=1VuMEp_YYNez0uFJfFi-bnWWQ2Gf1KLXSA-1x3IJLqR0), namely the fact that all content within the Pixelcash cloud is encrypted and an insane amount of data storage would be required to support a clone or hard fork. The total amount of data in the Pixelcash network gets significantly larger everyday and by the second, so even if a team of developers were to bypass the encryption and create a hard fork of the Pixelcash network, they would need to have an amount of storage space equal to the amount of data in the already existing network immediately available at the planned time of the fork; but if the team's calculations were off, and the amount of data in the network were to exceed the amount of storage space immediately on hand, the launch of the fork would be a failure. Such a large amount of data would also likely take a very long time to transfer into the new network, considering that the original network obtained the data bit by bit via mining over a long period of time, and thus the wait time until the new fork currency could be used would likely act as a deterrent.

CLOSURE

Pupa Pictura is Minarum Mille. If you think your cryptocurrency portfolio could use more .jpeg, then you should get your hands on some Pixelcash.

Kyle D. Usher