Kong: Sound Crypto Cash

The Kong Project
printer@kong.cash
Edition 1.0.1

The Kong project proposes a novel way to create sound physical cryptocurrency, or more appropriately crypto cash. There are two primary goals for Kong: (1) achieve a form factor that is conducive to circulation for the purchase of goods and services and (2) minimize the trust required in the issuer of crypto cash given the current state of the art in secure computing. Kong notes don't store cryptocurrency directly, but rather are backed by cryptocurrency in a trustless fashion through smart contracts. The holder of a Kong note can transparently audit – and ultimately access – the digital cryptocurrency based on a predefined set of conditions.

Contents

1 Introduction

1.1 Motivation

Bitcoin solved the problem of creating a viable digital currency through the use of cryptographic identities and proofs – the first decentralized "cryptocurrency". It pointed to a future where the money supply was not ordained by fiat, but rather achieved through a predefined set of rules and the consensus of its users.

It may be possible, however, that Bitcoin existing purely as a digital currency hampered its utility as a way to actually purchase goods and services. What if Bitcoin was tangible for its users from day one – what if, rather than attempting to explain Bitcoin to someone, you could hand it to them, fully formed in an embodiment that they could use like physical cash?

Paper notes and metal coins are intuitive to use; they indicate their denomination clearly and anyone who has used cash knows to secure them appropriately against theft or loss. Despite the rise of credit card payments, cash still accounts for 77% of all transactions globally.[1, 2] Physical notes and coins typically represent 5-10% of the value in fiat money supplies – the United States alone has issued cash instruments worth $1.74 trillion USD.[3, 4]

We posit the thesis that distributing a cryptocurrency primarily in a physical form may be beneficial to its broad adoption, purchasing power, and ultimately, its usage as a means of exchange for goods and services. Like Bitcoin, Kong is a means of exchange based on cryptographic proofs as opposed to absolute trust in financial intermediaries. Unlike Bitcoin, which was designed for commerce on the Internet, physical cryptocurrency is intended first for commerce in the real world. To achieve a physical cryptocurrency, however, we need to mirror the same trustless assumptions that underpin Bitcoin.

1.2 What is Kong?

Kong is a hybrid cryptocurrency that consists of both **physical notes** and **digital tokens**, where each of the former is backed by a fixed amount of the latter. Transactions in physical Kong notes are offline. They do not require any accompanying digital transaction to be valid and they are immune to factors that adversely effect on-chain transactions like spikes in transaction costs or network congestion. Transactions in physical Kong notes are *free*, *anonymous*, and *instantaneously final*.

Fig. 1. *Render of a 5 Kong note*

Kong notes are the first instance of **crypto cash**; sound, transferable credit backed by on-chain guarantees. They consist of a flexible circuit board, specialized secure element chips and an independent NFC interface capable of powering the secure element. Each secure element stores an internally-generated ECDSA key pair that is associated with a unique smart contract on the Ethereum blockchain. This smart contract escrows the amount of Kong tokens printed onto the note until a note-specific claim date (e.g. October 2022). Upon expiry, a cryptographic signature created by the note's secure element unlocks a function within the smart contract to transfer the digital token. Restrictions on the format of this signature guarantee that only the person who owns the physical note after expiry is capable of successfully calling the function.

The balance and code of each Kong note's associated smart contract is publicly visible and verifiable on the Ethereum blockchain. The coupling of a Kong note with a smart contract wallet is the first example of a **Si**licon-**Lo**cked Contract, or **SiLo**. Kong is strongly opinionated that crypto cash should be locked for an extended period of time – ideally years at a minimum – however, one can envision other assets which are immediately transferable from a SiLo or locked up permanently.

The security of physical Kong notes thus crucially depend on the security guarantees provided by the embedded hardware. Specifically, the security of each individual note is reliant on whether the marginal benefits of breaking the note's secure element (and extracting the key material) exceed the marginal costs. This marginal cost thus puts an upper bound on the value that each note can hold before attempts to break its security become economically worthwhile (see 2.4).

The Kong token is distributed in two novel ways: (1) minted onto physical Kong notes and (2) via a perpetual recurring lockdrop. A lockdrop is a means of distributing token by locking up a fungible crypto asset for a period of time – in the case of Kong, ETH. Kong's recurring lockdrop model is akin to mining rewards in other cryptocurrencies. Instead of the proof of work, however, the lockdrop represents the opportunity cost of how the participating ETH may have been otherwise invested, spent or sold. The Kong recurring lockdrop is elaborated in section 4.2.

2 Kong Notes

Kong notes require several important assumptions in order to function as sound crypto cash:

1. The ability to internally self-generate an asymmetric public-private key pair
2. The private portion of the key pair can be used to create proofs of authenticity which are verifiable against the public portion; in the case of an ECDSA public-private key this means signing data presented to the device
3. The private key is never revealed and cannot be derived at any point in the operation or life cycle of the device
4. The device should represent the simplest embodiment of a system capable of carrying out key generation and signing behavior as described in (1), (2) and (3), given the state of the art

In order to achieve these characteristics, Kong notes use a secure element chip. Unlike many secure elements available today, however, Kong notes require a variant with limited functionality and a published interface.

2.1 Form Factor

The Kong note includes many of the characteristics found in traditional paper currencies including a durable, flexible substrate, full color print indicating the denomination and claim date as well as novel visual security features. Like paper money, a Kong note can be folded and unfolded many times. The word elements for each Kong note are distributed across both the printed and copper layers. This paper does not detail all design and security elements in full, but instead focuses on those which allow for cryptographic verification of the note's authenticity.

Kong notes have been designed with an embedded NFC interface alongside the secure element, as shown in figure 2. This interface allows the holder of a Kong note to challenge the secure element via smartphone without any additional hardware. The NFC chip is isolated from the secure element and thus its removal or destruction does not impact the secure element or its ability to sign payloads required to verify and claim the Kong token. Likewise, attacks against the NFC chip do not impact the key material generated and stored by the secure element.

Fig. 2. *Front of the 100 Kong note*

Fig. 3. *Back of the 100 Kong note*

In the case of a damaged NFC antenna or chip, a Kong note can still be audited by one of three electrical interfaces which directly communicate with the I^2C interface on the secure element. This I^2C interface is a direct connection with the secure element chip and bypasses the NFC interface entirely. The backup connector shown in figure 3 lies directly beneath the secure element chip such that only a fragment of the note needs to remain intact for the token to be claimed.

2.2 Secure Elements

Secure elements are a special class of chip that are designed to generate and store cryptographic keys, as well as carry out basic cryptographic operations such as encryption and authentication. Some secure elements provide a full fledged instruction set like ARMv6 which allow for general computation. Broadly, secure elements can be considered to be a type of a HSM (or hardware security module), wherein all functions of the HSM are reduced to a single chip.

The goal of Kong is to maximize the cost of private key extraction relative to the state of the art in secure elements. Kong notes utilize a third party secure element chip with immutable key generation, storage, and signing functions that conform with the assumptions above. Any secure element wherein these functions are created or modifiable via a firmware update is not desirable as this creates additional layers of trust. Kong notes are the first instance of a cryptocurrency wallet where keys are generated and securely stored entirely within the confines of a non-programmable secure element chip and the wallet functions are openly auditable in a non-custodial smart contract. The printer of a Kong note never has access to this private material, nor can they duplicate it onto other notes.

An additional feature of the secure element embedded within Kong notes is the capability to sign data which is internal to the chip using an independent ECDSA key pair. This "internal" signing function is crucial as it can be used to generate signatures which attest to the nature of the chip – most importantly the fact the ECDSA key pair was self-generated, as well as signed random numbers which could have only originated from the chip itself.

2.3 Note on Smart Cards

Although marketed as a form of secure element, Java smart cards (or simply "smart cards") are not desirable for crypto cash for two key reasons:

1. Their cryptographic functions are typically specified in Java applets (small software programs) and they offer dangerous interfaces which allow for the extraction of private key material or modification of elliptic curve parameters by design.[5] Open applets for smart cards are a start towards increasing trustworthiness, however, they still require trust in the underlying smart card operating system.

2. Depending on make and model, smart cards host an array of proprietary crypto APIs that are largely undocumented and closed for open analysis. Frequently these APIs are still reliant on the security of the Java applet interfacing with them, and multiple vulnerabilities have been demonstrated against smart cards leveraging broken Java applet validation.[6, 7]

Other smart cards do not run the Java Card Platform, but instead rely on other proprietary operating systems. Regardless of platform, smart cards require trust both in the operating system as well as the programs carrying out cryptographic operations running on the smart card creating multiple software-level attack surfaces.

Kong notes do not use a secure element that allows for general programmability. The secure element in Kong notes hosts immutable cryptographic functions and thus has no additional firmware or operating system trust requirements. Figure 4 compares the additional layers of firmware required for crypto cash based on a Smart Card as opposed to Kong cash. The dashed line represents the boundary within which private key material would be available for computation.

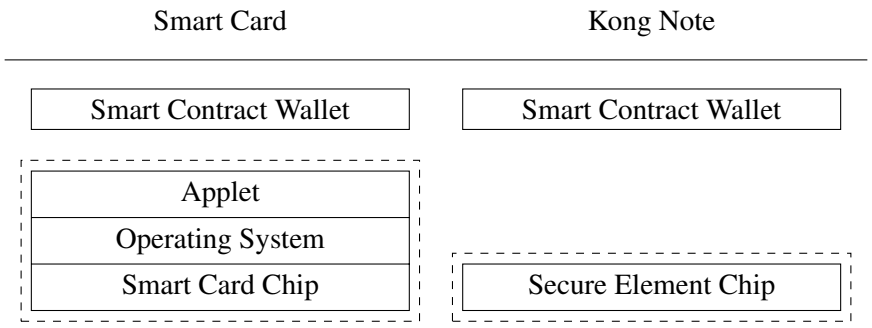

Fig. 4. *Comparison of a Smart Card wallet crypto cash stack vs. a Kong note*

2.4 Hardware Attacks

As noted in section 1.2, security guarantees of a given Kong note only extend as far as the costs required to extract a private key from the secure element chip. At the time of publishing, there are no demonstrated or known attacks against the secure element used within the Kong note, however, it's worthwhile to consider several kinds of attacks that might be possible:

- Fault Injection: deliberately introducing faults to the chip at an interface or electrical level such that the device leaks private key material through unintended operation. Fuzzing style attacks which reveal a hidden interface to extract private key material may require anywhere from $100 to 10,000+ USD of equipment. Inducing electrical faults may require costly equipment such as the NewAE ChipSHOUTER or actively modelocked infrared lasers.[8, 9]
- Side Channel Attacks: measuring the chip while it's performing a given operation, notably ones like encryption or signing. During these operations the chip might leak key material via detectable electromagnetic emissions. The cheapest attacks might be feasible with $100 USD of equipment ranging to thousands of dollars for sophisticated spectrum analyzers.
- Physical Probing: typically conducted by exposing the raw die ("decapping") such that optical or electrical probing can be carried out. The probing could be used to directly assess the state of the memory where the key is stored or extract any seed material used for generating entropy. Physical probing attacks typically require expensive equipment such as electron microscopes, ranging from $1,000 to 100,000+ USD.

It should be noted that these attack costs do not take into account the time or expertise required to carry out the attack. An attack requiring multiple hours per note could be signifcantly more expensive than the equipment costs noted above.

Even with the existence of a successful attack, attackers would still need physical access to any given note they wish to extract the key material from. Physical probing attacks like

chip decapping would significantly alter the look of the secure element chip, decreasing the ability of an attacker to re-circulate a compromised note.

See section 5.1 for further discussion on secure elements today and how their trustworthiness might be quantified.

3 Escrowing Token for Kong Notes

Physical Kong notes are backed by digital Kong tokens escrowed in smart contracts. See section 4 for more information on the Kong `Entropy` and `LockDrop` contracts which relate to the issuance of the Kong token.

For brevity, this description mostly focuses on interface and intended behavior rather than the full implementation. The complete set of Kong contracts, including additional documentation and tests, will be made available on Github.[10]

3.1 Goals and Design Philosophy

For each physical Kong note, one `Escrow` contract is deployed on the Ethereum blockchain. The purpose of each of these contracts is straightforward:

1. Hold digital Kong tokens in escrow for one specific Kong note.
2. Allow whoever holds the physical note to transfer it to an Ethereum account of choice once the escrow period has expired.
3. Attest to the origin and nature of the secure element embedded in the Kong note.

The `Escrow` and `Registry` contracts are designed to achieve these objectives under the restrictions imposed by the secure element, the public nature of the Ethereum blockchain, and security considerations.

3.2 Escrow Contract

Upon deployment, the `constructor()` function of `Escrow` sets the following variables, none of which are mutable after deployment:

- `uint256 publicKeyX;`
- `uint256 publicKeyY;`
- `address eccAddress;`
- `address tokAddress;`
- `address lockAmount;`
- `uint256 unlockTime;`
- `uint256 constant BLOCK_DELAY = 240;`

The first two variables, `publicKeyX` and `publicKeyY`, are the coordinates of the public key stored in the secure element in the physical Kong note. This public key is the primary characteristic that uniquely identifies each note.

The next two variables, `eccAddress` and `tokAddress`, set the addresses of two external contracts that each instance of `Escrow` interacts with. The first variable refers to the external `EllipticCurve` contract. Kong notes rely on ECDSA parameterized with curve secp256r1. Because Solidity provides no pre-implementation of secp256r1, we rely on a separate contract to verify secp256r1 signatures in Solidity, represented by `EllipticCurveInterface`. The second variable, `tokAddress`, refers to the Kong ERC20 token contract, represented by interface `KongERC20Interface`. `Escrow` calls this contract both to get its own Kong token balance and to transfer this balance to a new address (see below).

The constructor also sets the variable `unlockTime`, a UNIX timestamp that controls the date at which the function to transfer digital Kong tokens to another account is unlocked.

The constant `BLOCK_DELAY` controls how recent a signature must be to be acceptable to the `transferTokens()` function. We set this value to 240 so that `transferTokens()` only accepts block hashes from the last 240 blocks or about 1 hour at a block time of 15 seconds (240 * 15s = 3600s = 1h). The `BLOCK_DELAY` is inteded to ensure that only the most recent holder of a Kong note can claim the token off of it while considering factors like Ethereum network congestion.

3.3 Kong Registry

The Kong `Registry` contract identifies secure element chips usable for Kong notes. The `Registry` serves two purposes for Kong notes: resolution of the address of an associated `Escrow` contract for a given Kong note and an attestation of the model of a given secure element embedded in a Kong note.

The `Registry` stores several pieces of information about a given secure element in a Kong note. Importantly this includes sha256 hashes of the secure element manufacturer, model, manufacturer serial number and associated ECDSA public key. The values stored are as follows:

- `bytes32 primaryPublicKeyHash;`
- `bytes32 secondaryPublicKeyHash;`
- `bytes32 tertiaryPublicKeyHash;`
- `address contractAddress;`
- `bytes32 hardwareManufacturer;`
- `bytes32 hardwareModel;`
- `bytes32 hardwareSerial;`
- `bytes32 hardwareConfig;`
- `uint256 kongAmount;`
- `bool mintable;`

From Registry, any holder of a Kong note can verify that it integrates a secure element which conforms to the assumptions discussed above. They can also look up the associated Escrow contract for a given Kong note to verify the face value matches the stored Kong token. The additional values stored in the Registry for each Kong note relating to Kong minting described in section 4.1.

3.4 Unlocking Digital Kong Tokens

The function transferTokens() takes the following arguments:

- address _to: The address to receive the contract's Kong balance.
- uint256 _block: The number of the block whose hash was signed by the secure element.
- uint256[2] calldata _rs: The r and s values of the signature.

When the function is called, the contract first checks three timing conditions:

```
require(block.timestamp >= unlockTime);
require(block.number >= _blockNumber);
require(block.number <= _blockNumber + BLOCK_DELAY);
```

The first condition ensures that transferTokens() can only be called after the claim date printed on a given note. The second and third condition limit the validity of block hashes to those of the previous 240 existing blocks.

Next, the contract verifies that the provided signature is valid:

```
require(_validateSignature(sha256(abi.encodePacked(_to,
    blockhash(_blockNumber))), _rs) == true);
```

The contract expects _rs to be a signature that signs the sha256 hash of the tightly packed arguments _to and blockhash(_blockNumber). We include the recipient's address as part of the expected signature format to ensure that _to is the recipient of the locked balance. By enforcing the age condition for the block hash and including it in the signature, we ensure that only the current owner of the note is capable of calling the function. The function called to verify the signature, _validateSignature(), is an internal function that calls the signature verification contract and returns true if the signature is valid for the public key stored in the contract.

Once all of these conditions are verified, the contract obtains its own Kong balance from the ERC20 contract and transfers the entire balance to _to.

4 Issuance

Kong crypto cash is an experiment which intends to test the thesis that a physical form factor for cryptocurrency can achieve the hallmarks of fiat currency like price stability against

goods and services as well as broad adoption by non-technical users. To test this question, Kong is the first cryptocurrency issued primarily in a physical, note form. At launch, Kong is distinct from both existing cryptocurrencies, which are not desirable to spend for goods and services due to speculative pressure, and stable tokens that are pinned to inflationary fiat currencies.

There are seven Kong denominations: 1, 5, 10, 20, 50, 100 and 500 Kong. These denominations are intended to mirror a range of typical fiat currency notes and allow for change to be made in a variety of ways depending on a given transaction.

A small amount of purely digital Kong is available via a recurring Kong lockdrop. The Kong lockdrop grows the total Kong by up to 1% annually. The digital Kong emitted via lockdrop compensates for the significant amount of Kong notes that will be destroyed prior to the claim date. Any token associated with a destroyed note is permanently locked and therefore no longer in circulation. The recurring lockdrops also ensure that some digital Kong will be available for development purposes prior to the first claim date; for instance to integrate Kong in a decentralized finance application. Section 4.2 details how the recurring Kong lockdrop functions.

4.1 Minting Kong

At the point of manufacture, `Escrow` contracts associated with Kong notes are unloaded. The initial holder of a Kong must carry out a minting function against the note which consists of generating a random number leveraging the entropy present in the secure element. `Registry` stores three variables which control the availability and functionality of this minting function:

- `bytes32 tertiaryPublicKeyHash`: A sha256 hash of the provisioning key pair.
- `uint256 kongAmount`: The face value of a given Kong note.
- `bool mintable`: Boolean determining whether or not the note is mintable.

If the note is `mintable`, then the holder of the Kong note may request that the secure element generate a random number and sign it using the ECDSA provisioning key pair. The provisioning ECDSA key pair is capable of signing only data internal to the secure element and cannot be presented with an external string. An increment-only counter within the secure element may limit the number of times that the provisioning key pair can be used to generate signatures.

This random number and its corresponding signature are then submitted to the `Entropy` contract via the function `submitEntropy()` with the following parameters:

- `bytes32 primaryPublicKeyHash`: The sha256 hash of the primary public key.
- `uint256 tertiaryPublicKeyX`: The x-coordinate of the tertiary public key.
- `uint256 tertiaryPublicKeyY`: The y-coordinate of the tertiary public key.
- `bytes32 messageHash`: The hash of the signed message.
- `uint256[2] memory rs`: The array containing the r & s values of the signature.

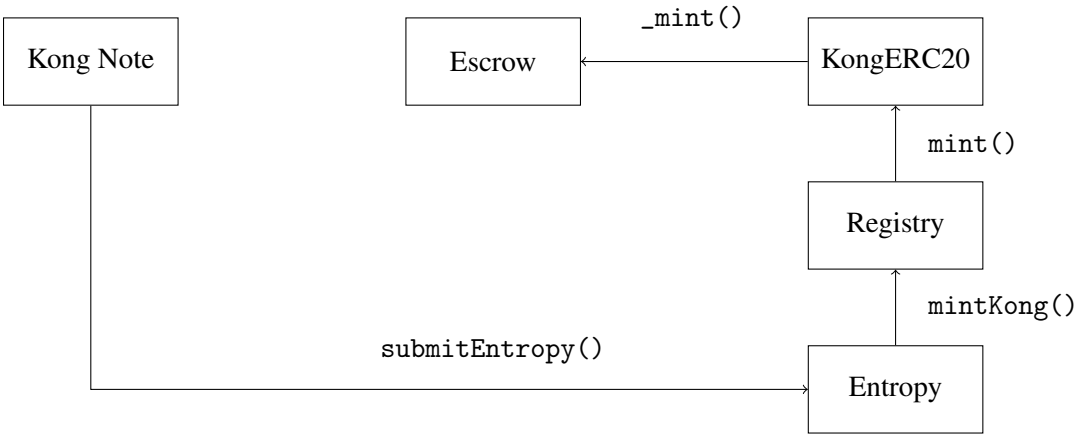

Fig. 5. *Minting Kong through submission of random number signed by provisioing key*

Depending on the denomination of the Kong note according to `Registry`, a difficulty adjustment in `Entropy` varies the gas cost of submitting the random number and signature.

If the signature verifies against the random number and the provisioning key, `Entropy` calls the function `mintKong()` within the `Registry` contract. If the Kong note is `mintable`, the `Registry` function in turn calls `mint()` in `KongERC20`. `KongERC20` verifies that the token is available within minting limits and mints token matching the face value of the Kong note into the corresponding `Escrow` contract via the standard ERC20 `_mint()` function.

4.2 Recurring Lockdrop Contract

A lockdrop is a means of distributing token to users which demonstrate that they have assumed the opportunity cost of locking up a crypto asset for a period of time. At the end of the lockdrop period they can then claim back their locked crypto asset as well as a portion of the token in a given lockdrop.

Lockdrops thus far have been singular token issuance events[11]; Kong is available via the first perpetually recurring lockdrop contract – one lockdrop can be initiated once every 30 days. The recurring nature of the Kong lockdrop is designed to mirror mining rewards; users must incur an opportunity cost in order to claim a portion of Kong token in a similar fashion to the way in which miners must incur costs to compete for block rewards in proof of work mining schemes.

Kong is distributed proportionally to those accounts which stake ETH into a given `LockDrop` instance based on two factors: the amount of ETH staked and the period of time that ETH is staked for, between 30 and 365 days. These two factors equally influence the amount of Kong that will be claimable at the end of a `LockDrop` instance.

The Kong lockdrop consists of several functions:

- `beginLockDrop()`: Located in `KongERC20`, this function will allow any account to initiate an instance of `LockDrop` as long as at least 30 days has past since creation of

the previous lockdrop. An amount of token representing approximately 1/12 of 1% (0.08295381%) of the total supply of Kong token, `MINING_REWARD`, is minted into the `LockDrop` instance. The first account to successfully call the function after the completion of the previous lockdrop deposit period will receive 256 Kong token.

- `stakeETH()`: Once a `LockDrop` contract has been deployed by `beginLockDrop()`, `stakeETH()` can be called to participate in that instance of `LockDrop`. It includes two parameters, the value to be staked in ETH and the staking period, `uint256 stakingPeriod`. All value staked is immediately sent to an individual instance of `LockETH`.
- `claimKong()`: At the completion of the staking period this function can be called against `LockDrop` to claim the proportional amount of Kong owed.
- `unlockETH()`: Present in each instance of `LockETH`, this function releases the staked ETH at the end of the lockdrop period. The ETH will be returned to the address which deposited it.

The Kong lockdrop does not contain any bonus or "signaling" functions. Users must incur a direct opportunity cost through the locking of ETH in order to participate in the lockdrop.

4.3 Example Kong Lockdrop

Juanita initiates a Kong `LockDrop` by calling `beginLockDrop()`. At that moment, 12,054,902 Kong token have been minted across Kong notes, the genesis token and previous lockdrops. She receives 256 Kong token. The `LockDrop` instance she created is available for staking as follows:

1. At the point at which the lockdrop is initiated, approximately 1/12 of 1% of the outstanding Kong token, or 10,000 Kong, are minted into the lockdrop contract. This is the `MINTING_REWARD`.
2. For a period of 30 days any account can lock ETH through the `LockDrop` instance by calling `stakeETH()`.
3. Each account can only participate in a given `LockDrop` instance with a single stake.

There are two participants, Lee and Enzo, who decide to lock ETH on the last day the `LockDrop` instance is open for staking:

- Lee sets a `stakingPeriod` of 30 days and locks 10 ETH.
- Enzo sets a `stakingPeriod` of 60 days and locks 5 ETH.

Lee and Enzo's weights in the lockdrop can be calculated by multiplying each of their `stakingPeriod`'s by amount of ETH each staked. In this case that is 10 * 30, or 300 for Lee, and 5 * 60, also 300 for Enzo. Each participant will receives a portion of the total Kong token in `LockDrop` in relation to their weights; 300/600 or 50% for Lee and 300/600 or 50% for Enzo.

After his `stakingPeriod` of 30 days, Lee can call `claimKong()` to receive his proportion of Kong token from the `LockDrop` instance, 6,000 Kong. Lee may also call `unlockETH()` which will release his ETH to the account he used to deposit it.

Because Enzo staked for 60 days, he must wait another 30 days until his `stakingPeriod` of 60 days is complete at which point he can claim his proportion of the `LockDrop` instance, also 6,000 Kong. Likewise, he can also call `unlockETH()` at this point in time to release his ETH.

Had either Lee or Enzo locked up ETH at the beginning of the lockdrop period as opposed to the end, then they would have been compensated for that time as well.

4.4 Kong Supply

For the first four years after the genesis of Kong, the following amount of Kong can be issued in a physical note form through the minting process detailed in section 4.1:

- **Year 1**: 2 ** 25 or 33,554,432 Kong
- **Year 2**: 2 ** 24 or 16,777,216 Kong
- **Year 3**: 2 ** 23 or 8,388,608 Kong
- **Year 4**: 2 ** 22 or 4,194,304 Kong

An additional 7,340,032 Kong was issued to the Kong project for discretionary Kong note printing and distribution. As detailed above, each Kong lockdrop increases the Kong supply by approximately 1% annually based on the total supply at the time the lockdrop is started. The frequency at which lockdrops are initiated is dependent on external accounts calling the `beginLockDrop()` function.

The upper bound for Kong token created after five years (the point at which only the recurring lockdrop rewards remain) is roughly 72,700,000 Kong, although this is highly dependent on how many Kong notes are ultimately printed and when lockdrops are created. It's also possible that there will be no increase in the Kong token supply if no account calls the lockdrop function.

The Kong token was deployed August 9th, 2019 at the address `0x177f2ace25f81fc50f9f6e9193adf5ac758e8098`. The first instance of `LockDrop` took place on the same day.

5 Future Research

5.1 ARX and Trustworthy Silicon

Kong's approach to crypto cash relies on reducing trust required in a given Kong note to a single secure element chip. This is a radical advancement in trust minimzation as it removes firmware from the system, however, as detailed in section 2, the trustworthiness of Kong is then reliant upon the state of the art in secure elements. There are several secure elements available on the market today which conform to the features required for a Kong note, however, silicon manufacturers do not broadly attest to the manner in which their chips were fabricated.

Instead, vendors typically opt to make claims about the chips through Common Criteria standards indicating their Evaluation Assurance Level (EAL) which are meant to indicate the kind of verification process a piece of hardware has been through. Unfortunately these standards are ambiguous for end users — an EAL 7 product is not necessarily more or less secure than an EAL 3 product, but it does, in theory, indicate that it went through a more thorough examination. Most importantly, however, these standards give little indication to the nature of the features of a given chip. An EAL 7 certified chip generating entropy from a single seed known by the manufacturer provides objectively worse randomness than an uncertified chip generating entropy from a physically unclonable function.

If there is significant demand for Kong notes or SiLos, then it follows that the broader decentralized web requires a new set of open secure element chips that conform to the assumptions laid out in section 2.

More desirable would be to devise a means of incentives that seek to quantify the security of a given secure element on economic terms as opposed to subjective claims. The ARX project is concerned with exploring this topic further; it's founded on the belief that no secure element will ultimately be unhackable, but that the state of the art in secure silicon can be continually advanced with incentives available to reveal attack vectors.[12] Ultimately a secure element with these incentives in place may be able to freely participate in the `Registry`.

5.2 Alternative Proofs

Kong is based on public-private key cryptography through the use of secp256r1 ECDSA signatures to attest to the presence of a private key. Given a different model of secure element chip, it would be possible to create a SiLo leveraging a different set of curve parameters like secp256k1. An entirely different public-private scheme would be feasible as well.

In the same way that a number of cryptocurrency projects are exploring zero knowledge proofs as a means of asserting membership in a group, so too could SiLos with proprietary secure elements support these schemes in lieu of secp256r1. More work needs to be done in this domain to understand the chip complexity and power requirements for a design based on a zero knowledge proof.

Another avenue of exploration is changing the means by which Kong notes indicate they are "valid" with respect to a crypto asset by using a completely different construction than registering a public key in the smart contract. One approach might involve Merkle trees wherein a given note can be demonstrated to be part of a large set of notes.

5.3 Re-locking Kong

The initial run of Kong notes was created with a claim date three years in the future (October 2022). This lock period was chosen in order to balance the potential for Kong notes to wear and the likelihood that a low cost attack vector will be discovered against the secure element.

The wear of existing paper currencies was also considered, with some notes having a lifespan of less than three years and others exceeding ten years[13, 14].

Given improvements in durability and secure element trustworthiness, it may be possible that Kong notes could easily circulate for a decade or more. In that case, it might be desirable to allow the holder of a Kong note to re-lock the digital Kong to the note for an additional period of time. The primary challenge that would need to be overcome would be the ambiguity of the new claim date with respect to the printed claim date. Without a means of modifying the claim date, someone may be reluctant to accept a re-locked Kong note.

One approach may be through physical alteration of the note in a fashion apparent to the secure element. Ideally this modification could be recorded on chain in order to assess the veracity of an extended claim date. Additional research is required to understand the likelihood of potential attack vectors and social engineering attacks to re-locked Kong.

6 Outlook

The Kong project seeks to challenge the nature of cryptocurrencies as abstract, digital-only constructs through the use of novel smart contracts, secure element chips, and token minting mechanisms. The result is a non-custodial physical cryptocurrency that embodies all the psychological touch points of cash that cross cultural and societal boundaries so that anyone can realize cryptocurrency without a prior understanding of elliptic curve math, private keys or addresses.

The first iteration of Kong is strongly opinionated with respect to mirroring the functionality of cash closely, however, it's possible to envision that new financial instruments designed around the same core principles of Kong with distinct operating models may emerge.

At the time of publishing, over 6,000 Kong notes have been created, 2,500 of which have been loaded with Kong token.

A special thanks to Kelsie, Chris Robison, barrywhitehat, Eric Meltzer, Althea Allen and Tyler Spalding for feedback on early drafts of this paper.

References

[1] Global payments 2018: A dynamic industry continues to break new ground. https://www.mckinsey.com/~/media/McKinsey/Industries/Financial%20Services/Our%20Insights/Global%20payments%20Expansive%20growth%20targeted%20opportunities/Global-payments-map-2018.ashx, Oct 2018.

[2] High internet use and state support help countries ditch cash. https://www.economist.com/graphic-detail/2019/08/01/high-internet-use-and-state-support-help-countries-ditch-cash, Aug 2019.

[3] The Fed - How much U.S. currency is in circulation? https://www.federalreserve.gov/faqs/currency_12773.htm, Nov 2019.

[4] Shaun O'Brien Raynil Kumar, Tayeba Maktabi. 2018 Findings from the Diary of Consumer Payment Choice. https://www.frbsf.org/cash/files/federal-reserve-cpo-2018-diary-of-consumer-payment-choice-110118.pdf, Nov 2018.

[5] DSAPrivatekey (Java Card API, Classic Edition). https://docs.oracle.com/javacard/3.0.5/api/javacard/security/DSAPrivateKey.html.

[6] Sergei Volokitin. Good, bad and ugly design of java card security. https://www.ru.nl/publish/pages/769526/sergei_volokitin.pdf, Jun 2016.

[7] Jean-Louis Lanet Guillaume Bouffard. Reversing the operating system of a java based smart card. https://www.researchgate.net/publication/262604854_Reversing_the_Operating_System_of_a_Java_Based_Smart_Card, Jul 2014.

[8] ChipSHOUTER® Kit. http://store.newae.com/chipshouter-kit/.

[9] Federico Menarini Jasper G. J. van Woudenberg, Marc F. Witteman. Practical optical fault injection on secure microcontrollers. https://www.riscure.com/uploads/2017/09/Practical-optical-fault-injection-on-secure-microcontrollers.pdf.

[10] Kong Github. https://github.com/kong-org.

[11] Edgeware Lockdrop. https://edgewa.re/lockdrop/.

[12] The ARX Project. https://arx.org.

[13] The Fed - How long is the lifespan of U.S. paper money? https://www.federalreserve.gov/faqs/how-long-is-the-life-span-of-us-paper-money.htm, Mar 2017.

[14] PE INTERNATIONAL AG commissioned by The Bank of England. LCA of Paper and Polymer Bank Notes. https://www.bankofengland.co.uk/-/media/boe/files/banknotes/polymer/lca-of-paper-and-polymer-bank-notes.pdf, Jun 2013.