namecoin

# Namecoin: Decentralized DNS-Like Identifiers

Jeremy Rand
Lead Application Engineer, The Namecoin Project
https://www.namecoin.org/

jeremy@namecoin.org
OpenPGP: 5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85

Presented at ICANN58

# Full disclosure

- I'm one of the most active Namecoin developers.
- I'm unaware of any Namecoin developers who disagree with anything in this talk.

- However, I can't speak for all the developers about all things.

- (This talk was prepared in collaboration with Hugo Landau.)

# Underlying motivation of Namecoin

- Humans behave nondeterministically.
- Any system run by humans will behave nondeterministically

- Even if a system has ground rules that are supposed to be inviolable, ground rules that are enforced by humans will be inconsistently enforced.

- Human behavior in the distant future is even more nondeterministic.

- (H/t to Greg Maxwell's philosophical writings on this topic.)

# Underlying motivation of Namecoin (2)

- The DNS is (in large part) run by humans.
- Risk: the people involved in operating the DNS can behave nondeterministically.

- ICANN and the DNS can be subject to some political issues.

# Underlying motivation of Namecoin (3)

- Namecoin is an experiment to find out:
- Is it possible to build something vaguely similar to the DNS, but with as little involvement by humans as possible?
  - Thereby create a DNS-like system that behaves more deterministically than the DNS.
- The hope: such a system will be more reliable and more secure against some human-based failure modes, because of its determinism.

# Existing Identifier Systems:
# Manual naming at a site

- E.g. hosts file

✗ No global namespace; names only meaningful locally

✓ Safe from nondeterministic human third parties

✓ Human-meaningful names

# Existing Identifier Systems: Hierarchical naming

- E.g. DNS

✓ Global namespace
✗ Not safe from nondeterministic human third parties
✓ Human-meaningful names

- Good usability.
- Risky as root of trust.

# Existing Identifier Systems:
# The name is the hash

- Content addressing, e.g. BitTorrent

✓ Global namespace

✓ Safe from nondeterministic human third parties

✗ No human-meaningful names; content can never change

# Existing Identifier Systems:
# The name is the public key

- E.g. Tor's .onion services

✓ Global namespace
✓ Safe from nondeterministic human third parties
✗ No human-meaningful names; but content can change

- Safe as root of trust
- Poor usability: user sees https://idnxcnkne4qt76tg.onion

# Existing Identifier Systems:
# The name is the public key

- E.g. Tor's .onion services

✓ Global namespace

✓ Safe from nondeterministic human third parties

✗ No human-meaningful names; but content can change

- Safe as root of trust
- Poor usability: user sees ~~https://idnxcnkne4qt76tg.onion~~

  https://odmmeotgcfx65l5hn6ejkaruvai222vs7o7tmtllszqk5xbysola.onion

# Zooko's Triangle

- You may have noticed in the preceding slides …
  - 2x ✓
  - 1x ✗
- This is Zooko's Triangle.
  - Zooko Wilcox conjectured that it was impossible to achieve all 3.

# Append-only logs

- Append-only public logs are seeing increasing popularity to ensure accountability.
- The most successful example: Google's Certificate Transparency.
- Every single certificate being used on the web is being put into an append-only log.
- Eventually, browsers will probably require certificates to be logged to be valid.

- Even if you want to keep control over a system, you might want all actions to be published.

# Append-only logs:
# Certificate Transparency

- Certificate Transparency is an append only log for certificates.
- Who can write to the log? Anyone, but only certificates from recognized CA's can be written.
- This ensures logs don't get spammed with junk data.
- Manual list of trusted entities is cumbersome.

# Append-only logs: Namecoin

- Namecoin is an append-only log for name registrations and updates.
- Because Namecoin uses a blockchain, it prevents spam by imposing an economic cost to write data.
  - This cost is small but effective.
  - This disincentivises bad actors from mass-squatting on names, without relying on a manual list of trusted entities.

# Append-only logs:
# Namecoin

- ✓ Global namespace
- ✓ Safe from nondeterministic human third parties
- ✓ Human-meaningful names

- Solves Zooko's Triangle.

# Accountability via Namecoin

- Namecoin means that an append-only log for naming can be operated as an open forum, enhancing its utility.
- Accountability and transparency can be made a (cryptographically verifiable) public good.
- Independently of the system of rules that Namecoin uses for names, its nature as an append-only log means that if a bad actor does something, you always know.

# Hypothetical case study:
# An accountable root zone

- Accountability can satisfy otherwise suspicious parties that nothing's going on.
- Hypothetical example: Maintain the root zone as an append only log to satisfy countries worldwide that US control isn't being abused, even at the intergovernmental level.
- Root servers could feed directly from the log.
- A root zone maintained as an append-only log could satisfy countries that e.g. their ccTLD won't be interfered with for political reasons.

- Like seismic monitoring: used by countries to check on each other under the Nuclear Test Ban Treaty, securing peace.
- Trust, but verify.

# TLS Public Key Infrastructure

- The Certificate Authority system is problematic (even with Certificate Transparency).
  - Way too many nondeterministic humans involved who can make mistakes.
- DNSSEC/DANE could improve the situation.
  - But there are political issues: some people are nervous about the possibility of abuse of power by the DNS root or the TLD operators.
- Namecoin could provide the advantages of DNSSEC/DANE without the political problems.

# Namecoin and DNS

- We don't expect that most software, or even most name resolution libraries, will be aware of Namecoin.
- Instead, we expect that Namecoin-to-DNS bridge software will be installed locally, translating DNS queries into Namecoin queries and converting the Namecoin responses back into DNS.
- Namecoin uses the .bit TLD.
  - This is not registered with ICANN or IETF right now.
  - We'd like to find a workable way to register it, e.g. as a Special-Use Name (like .onion).

# Namecoin and DNS (2)

- Our reference implementation, called ncdns, acts like an authoritative DNS server for the .bit TLD (running on localhost).
- DNSSEC keys are generated at install time.
- We intentionally try to keep Namecoin's domain name spec easily mappable to DNS, so that bridge software can be easily used.

# Namecoin and DNS (3)

- Tell your recursive DNS server (e.g. Unbound) to use ncdns as authoritative for .bit, and supply it with ncdns's DNSSEC public key. (This is a few lines in unbound.conf.) In theory, everything *should* just work.
- In practice, some DNS features aren't widely supported (e.g. DANE), so we have to do some weird application customizations to make things work.
- I once was trying to keep track of how many different layers of witchcraft we were using to make Namecoin's DANE work properly in browsers that don't support DANE; I stopped counting at 5.

# Namecoin Use Case: Buying and Selling Names

- In DNS, buying or selling a name usually involves some counterparty risk or relying on an escrow agent.
- In Namecoin, the buyer and seller can jointly construct a transaction that atomically pays the seller and transfers the name to the buyer.
- This eliminates counterparty risk without requiring an escrow agent's services.

- (Implementation by Phelix.)

# Namecoin Use Case:
# Non-Interactive Buy/Sell Offers

- You can also create non-interactive buy or sell offers.

- Alice creates sell offer: "Willing to sell example.bit for 100 NMC."
- Alice signs sell offer with private key, proving that she owns example.bit and is willing to transfer it in exchange for 100 NMC.
- Alice posts the signed sell offer on a forum or pastebin.

- (Design by Ryan Castellucci)

# Namecoin Use Case: Non-Interactive Buy/Sell Offers (2)

- Bob sees the offer and wants to buy example.bit.
- Bob completes the offer by signing it with a private key that owns 100 NMC; the offer is now a valid Namecoin transaction.
- Bob can now broadcast the completed transaction to the Namecoin network without contacting Alice.

- (Design by Ryan Castellucci)

# Namecoin Use Case: Non-Interactive Buy/Sell Offers (3)

- Alice gets paid 100 NMC; Bob receives example.bit.
- The transaction is atomic; no counterparty risk and no escrow agent needed.
- This works for both buy offers and sell offers.

- The Namecoin protocol supports this use case; user-friendly tools hopefully coming soon.

- (Design by Ryan Castellucci)

# Namecoin Use Case: Multisig

- A name is usually owned by a single private key, but it can also be owned by M-of-N private keys.
- This can be a useful protection against a single compromised key.
- A board of directors could each have a private key, and updating the name might require a supermajority of the board.

- The Namecoin protocol supports this use case; user-friendly tools hopefully coming soon.

# Namecoin Multisig Use Case: Two-Factor Authentication

- Namecoin can allow very flexible name update policies to be built, depending on the security and UX needs of a name owner.
- For example, Alice is the owner of a name, but she wants to limit her risk of stolen private keys while not introducing too much counterparty risk, so she constructs the following policy:

- Alice contracts Trent to run a two-factor-authentication service.
- Alice can update her name with arbitrary data if Trent signs her update. (Trent promises to only do this after verifying via 2FA.)

- (Design based on GreenAddress in Bitcoin.)

# Namecoin Multisig Use Case: Two-Factor Authentication (2)

- Trent pre-signs a specific transaction to revoke Alice's primary TLSA record and gives the transaction to Alice. Alice can then revoke the TLSA record later by signing the transaction herself even if Trent is offline or maliciously refuses to sign at that time.
- Trent pre-signs a specific transaction to transfer the name to Alice's sole control, which is only valid X days in the future. Trent gives this transaction to Alice. Alice can then recover the name after X days even if Trent goes out of business or loses his private key.
- (Design based on GreenAddress in Bitcoin.)

# Namecoin Multisig Use Case: Two-Factor Authentication (3)

- Trent cannot transfer or update Alice's name without Alice's signature.
- Alice can verify that the pre-signed transactions are authentic and that she is protected from Trent, before she applies this policy to her name.

- These policies are specified in a scripting language and are enforced to the same level as standard signatures are.

- (Design based on GreenAddress in Bitcoin.)

# Namecoin Multisig Use Case: Two-Factor Authentication (4)

- Namecoin doesn't mean registrars go away: "registrars" in Namecoin might look like Trent.
- But Namecoin **does** mean that "registrars" have much less ability to harm their customers than in DNS. (Either accidental or malicious harm.)
- This might lead to decreased security budgets being necessary for registrars.
- (Design based on GreenAddress in Bitcoin.)

# DDoS Resistance

- DNS infrastructure has been targeted by recent DDoS attacks (e.g. the attack against Brian Krebs).
- Some people have suggested that Namecoin might be a useful defense.
- It's unclear exactly how well Namecoin would stand up to a DDoS attack.

# DDoS Resistance (2)

- However, the Bitcoin network has been subject to "stress tests" (DoS attack attempts) in the past few years.
- The stress tests were conducted by for-profit companies who had a financial incentive to make Bitcoin's network appear weak against DoS attacks.
- Bitcoin's network was pretty much unaffected by the stress tests.
- Would Namecoin fare just as well as Bitcoin did? Would attackers of Namecoin have similar resources as the Bitcoin stress testers? I don't know.

- I think it's an interesting candidate use case for Namecoin. More research would be interesting here.

# Tradeoffs: Malware

- Namecoin transactions are irreversible (a consequence of Namecoin being an append-only log).
- As a result, if a name is transferred to a new owner, the old owner can't get it back without the new owner's signature.

- This means that Namecoin names are somewhat more vulnerable to hostile takeover by malware.
- Human error by the name owner could also be a problem.

# Tradeoffs: Malware (2)

- Some workarounds to this include keeping private keys on an air-gapped machine, and/or assigning multisig or 2FA policies to names.

- This isn't necessarily all bad: I've heard security experts comment that one of the best public benefits of Bitcoin becoming popular is that people are finally taking endpoint security seriously.
- As Bitcoin becomes more mature, I think it is likely that endpoint security will improve substantially.

# Tradeoffs: Trademark Infringement

- Namecoin doesn't have a nondeterministic human to determine which name registrations are valid.
- This is why it has security benefits and is more resistant to political issues.

- However, that also means that if someone registers a name that infringes on a trademark, there's no way to disable that name registration.
- This is inherent to the definition of trademark infringement: determining whether infringement occurred requires a human, and Namecoin is designed to not be run by humans.

# Tradeoffs: Trademark Infringement (2)

- A workaround would be for users to opt into a list of known trademark-infringing names, which get blocked somewhere between the Namecoin client and the user's web browser.
- For example, the Namecoin-to-DNS bridge that the user has installed might support this as an option.
- Existing infrastructure for this already exists: PhishTank is an example.

# Tradeoffs: Trademark Infringement (3)

- Caveat: A user who *wants* to view a name that infringes on a trademark could intentionally disable the blocking.
- Since the purpose of trademark law is to avoid consumer confusion, this isn't really a problem – a user who does this probably knows what they're doing.

- Caveat: Someone could buy an infringing name solely to sell it to the trademark owner (squatting).
- Since registering names costs money, it is difficult for a single person to squat on a very large number of names (similarly to how DNS names costing money reduces squatting).

# Tradeoffs: Privacy

- Since the full set of Namecoin transactions is public, anyone can look at the transactions.
- Transaction graph analysis makes it fairly easy to figure out if two transactions were done by the same person.
- This also affects Bitcoin.

- So if you register two Namecoin names, it's probably a public record that both names were registered by the same person.
- And if you bought your namecoins from someone else, they can probably see what names you register.

# Tradeoffs: Privacy (2)

- A workaround is to purchase namecoins with a payment method that doesn't leave a public record. (I.e. don't use bitcoins to buy namecoins!)
- And use separate public/private keys for each name you purchase so that they aren't linkable in the transaction graph.

- Bank transfers may be a good way to buy namecoins without leaving a public record.
- Some experimental efforts exist to make Bitcoin-like currencies that have better privacy (e.g. Monero and Zcash); they have their own drawbacks but may be worthwhile to some users.

# Tradeoffs: Privacy (3)

- The reference implementation of Namecoin generally has poor privacy and makes it difficult to prevent the public from learning that all your names have common ownership.
- We plan to make improvements on this.

# Tradeoffs: Security of Append-Only Property

- All of the security properties of Namecoin are cryptographically verifiable, **with one major exception**.
- The protection of the **ordering** of Namecoin name operations is not cryptographically secure, but instead economically secure.
- It would cost a lot of money to re-order the name operations, and the further back in time you go, the more money it would cost.

- Namecoin usually assumes that ordering is probably immutable circa 2 hours after a name operation occurs.
- But this is a probabilistic and economic assumption; much weaker than relying solely on cryptography.

# Tradeoffs: Security of Append-Only Property (2)

- If you could re-order the transactions going back to when a name was registered, you could place a registration operation for that name before the legitimate registration, thus stealing the name.
- You could also re-order the name's renewal operations to occur after the expiration period, thus forcing the name to expire and allowing you to register it yourself.
- Neither of these attacks has ever happened in real life to Namecoin, but an increased adoption of Namecoin might increase motivation of attackers to attempt it.

# Tradeoffs: Security of Append-Only Property (3)

- Bitcoin has the same problem, but since Bitcoin's economy is much bigger than Namecoin's, Bitcoin gains additional security against such attacks.

- There is a lot of active research into solving the issue of secondary blockchains being less secure than Bitcoin, because a lot of improvements to Bitcoin (including some being pushed by well-funded companies) are more easily deployable if this is solved.
- We're keeping a close eye on this research area.

# Tradeoffs

- None of the workarounds I described for malware, trademarks, and privacy are as straightforward as the countermeasures taken by DNS.
- Finding more elegant fixes is an open research problem.

- However, for many use cases, these workarounds are sufficient.

# Direction of Development

- For average people, installing and using Namecoin is still relatively difficult.
- Especially if TLS is desired. (Which it should be.)
- We just received funding from the NLnet Foundation and the Internet Hardening Fund (with budget from the Netherlands Ministry of Economic Affairs); this funding will be used to improve usability and application support for Namecoin's usage as a TLS PKI.
  - Goal is that Namecoin integration with a computer's name resolution libraries and with major web browsers' TLS implementations will be installable in a single step (e.g. run a .exe installer on Windows, install a .deb package on Debian).
  - Also includes UX improvements for name owners, and scalability / performance improvements.
- This work is being done by Jeremy Rand, Hugo Landau, Brandon Roberts, and Joseph Bisch.

# Direction of Development (2)

- We're also engaging with The Tor Project.
- Tor's user base has specific security requirements that aren't well-suited to the DNS.
- They're using .onion now, which isn't human-meaningful (and this is going to get worse when their Onion Services v3 upgrade gets rolled out).
- Humans don't check the full .onion address, which means scammers are, in the wild, creating partial preimages of existing .onion addresses for impersonation.
- Tor is a good candidate for early adoption of Namecoin; they can probably live with the current state of Namecoin's tradeoffs because all the other available options don't meet Tor's security requirements.
- Jeremy Rand is leading outreach with The Tor Project.

# Direction of Development (3)

- On the backend, we have an upcoming "hardfork" (blockchain terminology for an upgrade that breaks backward-compatibility).
  - The hardfork was necessitated by some improvements in Bitcoin's codebase that we can't adopt without breaking backward compatibility.
- We're also investigating several other upgrades, such as:
  - Expiration period defined in real-world time rather than cryptographic approximation of real-world time.
  - Compact proofs of nonexistence.
  - Allowing Namecoin nodes to drop old data (hashes would be preserved, so the dropped data can still be proven).
  - Allowing namecoins to be purchased using bitcoins, without counterparty risk.
- Daniel Kraft is leading most of these backend efforts.

# Thanks for inviting me!

- Happy to take questions.

- https://www.namecoin.org/
- My email: jeremy@namecoin.org
- My OpenPGP:
  5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85