Posts Resources Other Bitcoin Sites Live Bitcoin Chat Live Price Chat

Join the discussion BECOME A REDDITOR

Posted by u/[deleted] 3 years ago

What's the difference between public key and public address?

So I've heard the two used interchangeably (incorrectly) From the technical papers I've read, it seems that you hash a public key to get your address. In order to receive Bitcoin, don't you just need to give someone a public key? Why go through that extra step to create a address, when you can give out your public key?

30 Comments Share ••• 89% Upvoted

This thread is archived
New comments cannot be posted and votes cannot be cast

SORT BY BEST

hksupport 32 points · 3 years ago

There's a few things to understand in order to answer that.

The first is what a hash is. The short version is that it's a function that returns a value of a specific length. So whether the input is "Hi" or "Hello" or "This is an even longer piece of data", the hash function will return the same length of output, and that output will always be the same size, and the output will always be the same for a given input. The other aspect of a hash function is that it is "one way". It is very easy to put input into the hash function and get output, but it's basically impossible to get some hash output and determine from that what the input was. The hash is considered broken and unusable if you are able to do that. The only way to map hash outputs to inputs is to just churn through inputs and outputs and record them in a lookup table. And hackers do that - they're called rainbow tables. That's why websites often add "salt" to something before they hash it. That "salt" is simply a small bit of extra arbitrary random data to add in, so that rainbow tables won't work. For example, your poor password

The next thing to know is about public and private keys. Whether it's elliptic curves or RSA, the general idea is the same. You have a "private key" from which you can derive your public key. People can encrypt things to your public key, and if they do so, then only your private key can decrypt them. So you can publish your public key publicly, and people can then encrypt messages that can only be unlocked with your private key. A second feature of private/public keys is digital signatures. A signature is something that can only be created by your private key, but which your public key can be used to verify that the signature indeed must have been created with your private key. So I can use my private key to "sign" a PDF document for example, and if you have my public key and the document and the signature text, you can verify that indeed I made that signature for that document, and I made it with my private key. The signature proves I own the private key, even though it doesn't reveal the private key to you or anybody else.

The most common way to send bitcoins is to an address, which is a hash of a Bitcoin public key. The reason we do it that way is so that if there is a vulnerability in elliptic curves, your money can still be safe, since the public key isn't even known until you spend the money, only the hash is known. The public key is revealed only when you spend money, because it is necessary to prove that the digital signature came from your private key. And the way Bitcoin verifies that the transaction is valid is it checks the signature with the public key (and the data being signed is not a PDF but the Bitcoin transaction itself), and the Bitcoin miners and users verify that the private key indeed did make that signature (and make it for this transaction), and then they also verify that the public key hash is the same hash as the output transaction. If those two things are true - the signature is good and the hash matches - then the transaction is valid and the money can be spent.

The reason Bitcoin uses the hash in addition to the public key is security. Let's say elliptic curves suffered a flaw that allowed somebody to derive a private key from a public key in 3 hours, but the hash functions were still unbroken. Then your money would be safe the vast majority of the time, because you could spend it and get the transaction mined before somebody looking at that transaction (which shows your signature and your public key) would be able to use your public key to reverse engineer your private key. And your public key wouldn't even be revealed until you spent the money, so as long as the hash function was still safe, you're mostly ok.

So it's a fail-safe backup. In an emergency where the elliptic curves or the hash function get broken, users still have an emergency backup way of keeping their money temporarily safe while the Bitcoin software gets updated. This is why a lot of people always say you should not re-use addresses, because if you've spent from an address previously, then its public key is already publicly known, and you lose that emergency backup protection.

So now that we've got all that explained, I can actually answer your question! An address is the hash of the public key. So when you have a private key, you can use that to derive the public key, which you can use to derive the address/hash. That's why you only need the private key backed up, because everything else can be derived from that.

When you hear about compressed/uncompressed, that has to do with a feature of the Bitcoin elliptic curve math. Basically, the public key is actually a point on a graph derived from the private key. It's just an X and Y coordinate. However, the Bitcoin curve has a neat property that how every positive X value can have a positive or negative Y value, because of the Y-squared part.)

So sometimes you'll see a public key written as "04" followed by the 64-character X value, followed by the 64-character Y value. That's an uncompressed public key. But you may also see just the X value with either an "02" in front or an "03" in front of it, indicating which of the two Y values is supposed to be used if you uncompressed it. So a compressed or uncompressed *address* is just saying that the address is the hash of either a longer 04 public key or of the shorter 02/03 public key. Remember, hashes output the same length, so the length of the address will be exactly the same, and in fact you won't know whether it's a compressed or uncompressed address until you see the spending transaction that shows the public key.

Share Report Save

BitcoinLibertarian 5 points · 3 years ago

Thank you for such a great explanation. Have a beer /u/changetip

Share Report Save

changetip 1 point · 3 years ago

The Bitcoin tip for a beer (12,153 bits/$3.50) has been collected by *hksupport*.

--

what is ^^ChangeTip?

Share Report Save

hksupport 1 point · 3 years ago

Thanks!

Share Report Save

fiat_sux4 2 points · 3 years ago

Wow, great explanation. One question though: What would be the advantages and disadvantages of using a compressed vs. uncompressed address? In other words, why bother having the two formats?

Share Report Save

murbul 1 point · 3 years ago

Uncompressed keys exist only because of historical reasons and backwards compatibility. It was either an oversight, or Satoshi was unaware of the existence of compressed keys when the first version was released. They were added in bitcoin-qt 0.6 (a long time ago) but a lot of wallets e.g. blockchain.info have lagged behind.

Compressed keys save 32 bytes per transaction input. It doesn't sound like much, but a lot of transactions are less than 300 bytes so it's a significant percentage saving with no cost.

The only disadvantage is so negligible it's barely worth mentioning. Working with compressed public keys requires an additional step to calculate the y-coordinate, but it's fiat_sux4 2 points · 3 years ago

Another great explanation. Thanks!

Share Report Save

Family_Shoe_Business 1 point · 3 years ago

This explanation is great. Thank you so much!

Share Report Save

kinoshitajona 25 points · 3 years ago

If ECDSA ever broke, you would still be protected by ripemd.

If ripemd also broke, you would still be protected by sha256.

If you spend bitcoins that were in an address even once, the ecdsa public key is published to the blockchain, so you lose the extra protection from ripemd and sha.

Share Report Save

Comment deleted by user 3 years ago

giszmo 0 points · 3 years ago

> Because the pubkey to address hashing can't be reversed, your pubkey is safe against (for example) quantum attacks.

Isn't that too short-sighted? Shouldn't brute-forcing a hash-reversal precisely be what quantum computers are good at? Maybe the step pubkey->privkey is the easier but I would expect the step from a hash to a valid pubkey should also be easier with a decent quantum computer.

Share Report Save

waxwing 12 points · 3 years ago · *edited 3 years ago*

> Shouldn't brute-forcing a hash-reversal precisely be what quantum computers are good at?

No, as it turns out, although it's not a simple story!

> However, the two algorithms differ drastically in just how efficient they are. Shor’s algorithm reduces the runtime of cracking elliptic curve cryptography from $O(2^{k/2})$ to $O(k^3)$ – that is to say, since Bitcoin private keys are 256 bits long, the number of computational steps needed to crack them goes down from 340 trillion trillion trillion to a few hundred million at most. Grover’s algorithm, on the other hand, does not offer anything close to so drastic a speedup. Rather, it simply provides a modest reduction from $O(2^k)$ to $O(2^{k/2})$. In the case of RIPEMD-160, the weaker of the two hashes used to create a Bitcoin address, this means that the number of steps needed to recover a public key from an address goes down from 1.4 trillion trillion trillion trillion to … 1.2 trillion trillion. Somewhat easier, but still thankfully impractical. As

> the bottleneck. “Unused” Bitcoin addresses, on the other hand, expose only the address itself, so it is the RIPEMD-160 Grover problem that poses the weakened, but still insurmountable, challenge.

From: https://bitcoinmagazine.com/6021/bitcoin-is-not-quantum-safe-and-how-we-can-fix/

I have no idea if this is anything close to the final word on the matter, but the whole article is worth a read if you're interested.

Share Report Save

Introshine 4 points · 3 years ago

http://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf

Share Report Save

[deleted] 2 points · 3 years ago

Not sure why you are being downvoted. If we're talking about a future where we have quantum computers that can solve discrete logarithms, then it's reasonable to think we could have quantum computers that can run Grover's algorithm on a domain size of $2^{160.}$

Share Report Save

Amichateur 6 points · 3 years ago

- The address is shorter than the pub key.
- Certain theoretical future cryptographic attack vectors like quantum computers are said to be possibly capable of deriving the private key from the public key, but not from the hash of the public key (from the address). So the sha256 provides considerable extra security.

Share Report Save

mrincompetent 1 point · 3 years ago

For some reason I thought the address was the same as the pub key, thanks for the info. So let me see if I get this now. There's a private key, from which you can derive a public key. Bitcoin can be sent to this public key, but usually an extra level of security is created by hashing the public key into an address which can also accept transactions, but from which you can not derive the public key.

Share Report Save

Amichateur 1 point · 3 years ago

Almost.

The transaction is never sent to a pub key but always to an address. There are many pub/private key pairs having the same address, each owner of such priv key can spend the coins.

Share Report Save

> From the technical papers I've read, it seems that you hash a public key to get your address.

That's true.

Addresses add features you don't get with a public key.

For starters, there is a checksum built into every address. Mistype or transpose a character, and it can be detected - before you (or your payer) send money down a hole.

Second, addresses avoid characters that are easily confused with each other through compact base 58 encoding. Hex doesn't have this problem, but results in very long representations.

Also, giving a hash of your public key offers a security advantage. If elliptic curve cryptography starts to succcumb to attacks in which a private key can be extracted from a public key (discrete log problem), using a public key hash forces the attacker to perform yet another difficult step of generating the public key from the hash. The added security only applies if you haven't reused addresses. Spending from an address publishes the corresponding public key to the block chain.

Some have speculated that Satoshi didn't know about public key compression at the time addresses were created. Uncompressed public keys are 65 bytes long, whereas the hash value Bitcoin uses is only 20 bytes long. From that perspective, using a hash leads to a more compact identifier. A compressed public key is 33 bytes long.

Share Report Save

luke-jr **Luke Dashjr - Bitcoin Expert** 4 points · 3 years ago

Public keys are a low-level primitive in the Bitcoin protocol, are (despite their name) actually *private information* and shouldn't ever be exposed to end users. Addresses are a high-level abstraction people send bitcoins to in order to transfer between wallets. They are implemented with scripts that quite often end up using public keys for authentication, but are entirely distinct from those keys. An address should only be given to the person you intend to pay you, who should only ever use it for a single payment.

Share Report Save

Petersurda 3 points · 3 years ago · *edited 3 years ago*

Even though the others here said that it's supposed to make bitcoin more secure (this is what I originally thought too), I recall one of the interviews of Andreas Antonopulous (or perhaps some other developer, Hearn or Andresen), where he said that Satoshi explained that he did this to shorten the signature (64 bytes public key length vs. 25 bytes address), and that had he more experience with cryptography he probably would have used Schnorr signatures instead to achieve the same goal.

Share Report Save

AussieCryptoCurrency 2 points · 3 years ago

A public key is the public key set of the private key.

pentarh 1 point · 3 years ago

Public key is a product of one-way ecdsa transformation of private key.

Address is a product of one-way hash160 transformation of public key.

Yes, you may give your pub key, it may not compromise your private key. Even more, when you make payment from your address - your public key is exposed in script sig of transaction.

But since you didn't make payments from Address, nobody knows your public key. It's additional level of privacy.

Share Report Save

CoinCadence 1 point · 3 years ago

- Uncompressed Public Key: 046591ed35e4a6c43684ede7b2cb37f6cc19fc6629a2abce8ee0f3e021e3f274e6013ba12219 4117982eb818322f3e3b12a7edb15d586cdff87714de281c4ce5ec
- Uncompressed Address: 19ZE9JY8NjX569nXGeHKwHzuqmaVvu3WuT
- Compressed Public Key: 026591ed35e4a6c43684ede7b2cb37f6cc19fc6629a2abce8ee0f3e021e3f274e6
- Compressed Address: 1DipSsz4YYnjufGqVkHcbjrJEyHnxsY3Ec

You can play with these at https://brainwallet.org/

The passphrase for the above key/address pairs is the thread title.

Share Report Save

chinawat 1 point · 3 years ago

It's unfortunate that such incorrect usages exist. The primary elements of Bitcoin that users should focus on are the "addresses" and "private keys". Even "public keys" are only secondary in importance for general users. There really is no such thing as a "public address" (or a "private address" for that matter) in the Bitcoin protocol.

Share Report Save

arichnad 1 point · 3 years ago

> In order to receive Bitcoin, don't you just need to give someone a public key?

No, this is the incorrect part that's confusing you.

In order to receive Bitcoin, you need to give someone a public **address**.

Share Report Save

Natanael_L 3 points · 3 years ago

Both works

True, but a user should never give out a public key that hasn't already been published. Addresses are intended to be distributed.

Share Report Save

cointastical 2 points · 3 years ago

You can derive a Bitcoin address from the public key.

Share Report Save

chinawat 1 point · 3 years ago

But we shouldn't give the impression that giving out either one would have the same effect. Giving out an address is normal best practice. There's generally no need to ever reveal a public key.

Share Report Save

toliro 0 points · 3 years ago · *edited 3 years ago*

In Bitcoin there are private keys and public keys just like in PGP.

A public key is also known as a bitcoin address.

Thus, all bitcoin addresses are public.

So "public address" is actually redundant.

Public keys are the receiving point. Public keys are the sending point. The public key is derived from the private key. The coins are stored in both of the keys, but only the private key will allow you to sign the transaction to send (spend) the coins.

Share Report Save

murbul 2 points · 3 years ago

Wrong. Read any of the other replies here to see why.

Share Report Save