



c++ code to generate bitcoin wallet key pairs

Details Proposals

Project Details

$30.00 – 250.00 USD

BIDDING ENDS IN 5 DAYS, 23 HOURS

I need c++ code to generate bitcoin wallet key pairs.
I Would like to program to run and generate 20 public and private key pairs to a text file.
You will need to read up on how to do this but basically it involves a 10 step program.

How to create Bitcoin Address
1 - Having a private ECDSA key (elliptic curve https://www.youtube.com/watch?v=dCvB-mhkT0w or read detailed article https://devcentral.f5.com/articles/real-cryptography-has-curves-making-the-case-for-ecc-20832 or see note on "Elliptic Curve Cryptography") (elliptic curve = y^2 = x^3+ax+b) for bitcoin y2=x3+0x+7 (a=0 and b=7) also https://www.youtube.com/watch?v=iB3HcPgm_FI

for bitcoin elliptic curve y^2=x^3 + 0x+7

see graph on https://www.geogebra.org/graphing/n2wmv8uq
18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725
AS A NUMBER
1125356301205968582595361922210782354909214769903167223838579036935154264246 9 (converted with https://www.rapidtables.com/convert/number/hex-to-decimal.html)

2 - Take the corresponding public key generated with it (65 bytes, 1 byte 0x04, 32 bytes corresponding to X coordinate, 32 bytes corresponding to Y coordinate)
0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6 (THIS IS A BINARY NUMBER GO HERE http://www.fileformat.info/tool/hash.htm?hex=600ffe422b4e00731a59557a5cca46cc183944191006324a447bdb2d98d4b408 THEN ENTER THIS NUMBER THE OUTPUT WILL SHOW THE NUMBER BELOW UNDER SHA256)
NUMBER
57848633364904088560474730487813481066629663012816365002383147711370517113092174008031862913129373759877648374910375401539469261706206018832290100939336614

3 - Perform SHA-256 hashing on the public key

600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B408

4 - Perform RIPEMD-160 hashing on the result of SHA-256
010966776006953D5567439E5E39F86A0D273BEE

5 - Add version byte in front of RIPEMD-160 hash (0x00 for Main Network)
00010966776006953D5567439E5E39F86A0D273BEE
(note that below steps are the Base58Check encoding, which has multiple library options available implementing it)

6 - Perform SHA-256 hash on the extended RIPEMD-160 result
445C7A8007A93D8733188288BB320A8FE2DEBD2AE1B47F0F50BC10BAE845C094

7 - Perform SHA-256 hash on the result of the previous SHA-256 hash
D61967F63C7DD183914A4AE452C9F6AD5D462CE3D277798075B107615C1A8A30

8 - Take the first 4 bytes of the second SHA-256 hash. This is the address checksum
D61967F6

9 - Add the 4 checksum bytes from stage 7 at the end of extended RIPEMD-160 hash from stage 4. This is the 25-byte binary Bitcoin Address.
00010966776006953D5567439E5E39F86A0D273BEED61967F6

10 - Convert the result from a byte string into a base58 string using Base58Check encoding. This is the most commonly used Bitcoin Address format
16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM

Skills Required

PHP | C Programming | Algorithm | Software Architecture | C++ Programming

Project ID: 20537760

Complete your profile

Please complete these 3 steps before bidding on the project

Update your skills

Verify your email

Update your profile

You must add at least one of these skills to bid on this project.

- [ ] PHP
- [ ] C Programming
- [ ] Algorithm
- [ ] Software Architecture
- [ ] C++ Programming

Select All Next

About the Employer

South Africa

25 projects completed

5.0 (17 reviews)

Member since Jul 25, 2008

Employer Verification

Payment method verified

Deposit made

Email address verified

Profile completed

Phone number verified

Bids Left	8/8
Until next bid	-
Refresh speed	1x
Average bid	$192 USD

Bookmark Project

Report Project

US (International) / English

Help & Support

Freelancer

Categories
Projects
Contests
Freelancers
Enterprise
Preferred Freelancer Program
Project Management
Local Jobs
Showcase
API for Developers

About

About us
How it Works
Security
Investor
Sitemap
Quotes
News

Terms

Privacy Policy
Terms and Conditions
Copyright Policy

Code of Conduct
Fees and Charges

Apps