ബിറ്റ്കോയിൻ: പരസ്പര പണവിനിമയത്തിനുള്ള ഒരു ഇലക്ട്രോണിക് സംവിധാനം

സതോഷി നകാമോട്ടോ satoshin@gmx.com www.bitcoin.or

സംഗ്രഹം.

ഇലക്ട്രോണിക് പണത്തിന്റെ പിയർ-ടു-പിയർ, ഓൺലൈൻ പണമിടപാടുകൾ ഒരു ധനകാര്യ സ്ഥാപനത്തിലൂടെയല്ലാതെ ഒരു കക്ഷിയിൽ നിന്ന് മറ്റൊരാളിലേക്ക് നേരിട്ട് അയയ്ക്കാനുള്ള സംവിധാനം ഒരുങ്ങുന്നു. ഡിജിറ്റൽ ഒപ്പുകൾ ഒരു പരിധിവരെ പരിഹാരം നൽകുമ്പോഴും രണ്ടുതവണയുണ്ടാകുന്ന ചെലവ് തടയാൻ ഒരു മൂന്നാംകക്ഷി ഇല്ലെങ്കിൽ പ്രധാനപ്പെട്ട ആനുകൂല്യങ്ങൾ നഷ്ടപ്പെട്ടേക്കാം. ഒരു പിയർ-ടു-പിയർ നെറ്റ്വർക്ക് ഉപയോഗിക്കുക വഴി രണ്ടുതവണയുണ്ടാകുന്ന ചെലവ് പ്രശ്നത്തിന് ഞങ്ങൾ ഒരു പരിഹാരം നിർദ്ദേശിക്കുന്നു. ഇത് പ്രൂഫ് ഓഫ് വർക്ക് വീണ്ടും ചെയ്യാതെ സ്ഥിരമായ ഒരു റെക്കോർഡ് സൃഷ്ടിക്കുന്നതിനാൽ ഹാഷ് അടിസ്ഥാനമാക്കിയുള്ള പ്രൂഫ് ഓഫ് വർക്ക് ശൃംഖലയിലേക്ക് ഹാഷ് ചെയ്തുകൊണ്ട് നെറ്റ്വർക്ക് ഇടപാടുകൾ ടൈംസ്റ്റാമ്പ് ചെയ്യുന്നു. ഏറ്റവും ദൈർഘ്യമേറിയ ശൃംഖല, സാക്ഷ്യം വഹിച്ച ഇവന്റുകളുടെ ക്രമത്തിന്റെ തെളിവെന്ന നിലയിൽ മാത്രമല്ല, സിപിയു ശക്തിയുടെ ഏറ്റവും വലിയ പൂളിൽ നിന്നാണ് വന്നതെന്നതിന്റെ തെളിവായും പ്രവർത്തിക്കുന്നു. നെറ്റ്വർക്കിനെ ആക്രമിക്കുന്നത് തടയുന്ന നോഡുകളാണ് സിപിയു ശക്തിയുടെ ഭൂരിഭാഗവും നിയന്ത്രിക്കുന്നത്. അവ ഏറ്റവും ദൈർഘ്യമേറിയ ശൃംഖല. സൃഷ്ടിക്കുകയും ആക്രമണങ്ങളെ അതിജീവിക്കുകയും ചെയ്യുന്നു. നെറ്റ്വർക്കിന് മിനിമം ഘടന ആവശ്യമാണ്. സന്ദേശങ്ങൾ മികച്ച രീതിയിൽ പ്രക്ഷേപണം ചെയ്യുന്നത്, കൂടാതെ നോഡുകൾക്ക് ഇഷ്ടാനുസരണം നെറ്റ്വർക്കിൽ നിന്ന് പുറത്തുകടക്കാനും തിരിച്ചു വരാനും കഴിയും എന്ന പ്രത്യേകതയുമുണ്ട്, നോഡുകൾ പോയപ്പോഴുണ്ടായ ഫലങ്ങളുടെ തെളിവായി ഏറ്റവും ദൈർഘ്യമേറിയ പ്രൂഫ്-ഓഫ്-വർക്ക് ശൃംഖല സ്വീകരിക്കുന്നു.

1. ആമുഖം

ഇന്റർനെറ്റിലെ വാണിജ്യഇടപാടുകളുടെ ഭാഗമായി, ഇലക്ട്രോണിക് പണമിടപാടുകൾ നടത്തുന്നതിന് പൊതുവിൽ വിശ്വസനീയമായ മൂന്നാം കക്ഷികളായി പ്രവർത്തിക്കുന്ന ധനകാര്യ സ്ഥാപനങ്ങളെ മാത്രമാണ് നിലവിൽ ആശ്രയിക്കുന്നത്. മിക്ക ഇടപാടുകൾക്കും സിസ്റ്റം നന്നായി പ്രവർത്തിക്കുന്നുണ്ടെങ്കിലും, ട്രസ്റ്റ് അധിഷ്ഠിത മോഡലിന്റെ സ്വാഭാവികമായ ബലഹീനതകൾ ഇപ്പോഴുമുണ്ട്. ധനകാര്യ സ്ഥാപനങ്ങളെ സംബന്ധിച്ച് മധ്യസ്ഥതല തർക്കങ്ങൾ ഒഴിവാക്കാൻ കഴിയാത്തതിനാൽ

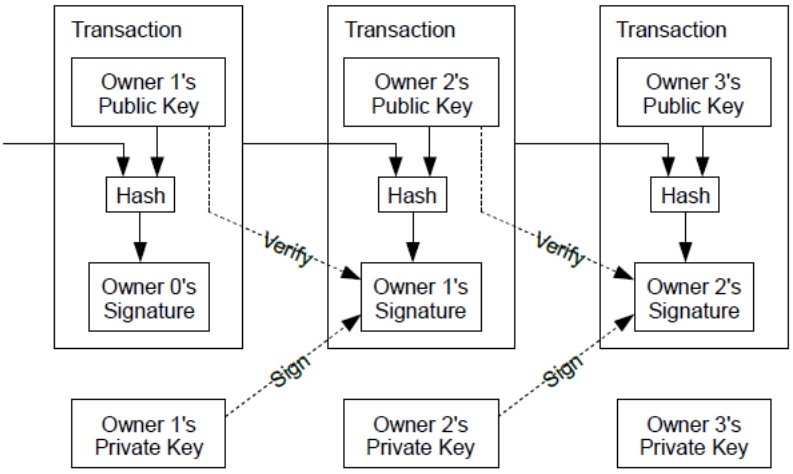
പൂർണ്ണമായും തിരിച്ചെടുക്കാനാകാത്ത ഇടപാടുകൾ യഥാർത്ഥത്തിൽ സാധ്യമല്ല. ഇത് മധ്യസ്ഥതയുടെ ചെലവ് വർദ്ധിപ്പിക്കുകന്നതിനൊപ്പം മിനിമം ഇടപാടിന്റെ അളവ് പരിമിതപ്പെടുത്തുകയും ചെറിയ ഇടപാടുകൾക്കുള്ള സാധ്യത കുറയ്ക്കുകയും ചെയ്യുന്നു, ഇത് തിരിച്ചുനൽകാനാകാത്ത സേവനങ്ങൾക്കായി തിരിച്ചുനൽകുന്ന പണമിടപാടുകൾ നടത്താനുള്ള കഴിവ് നഷ്ടപ്പെടുന്നതിനാൽ ചെലവ് ഗണ്യമായി വർധിപ്പിക്കുന്നു. ഇത് വിശ്വാസത്തിന്റെ ആവശ്യകത വർധിപ്പിക്കുന്നു. വ്യാപാരികൾ തങ്ങളുടെ ഉപഭോക്താക്കളെക്കുറിച്ച് ജാഗ്രത പാലിക്കേണ്ടതുണ്ട്, ആവശ്യമുള്ളതിനേക്കാൾ കൂടുതൽ വിവരങ്ങൾ അവരിൽ നിന്ന് തേടണം. എന്നിരുന്നാലും ചെറിയൊരളവ് കബളിപ്പിക്കൽ ഈ മേഖലയിൽ ഒഴിവാക്കാനാകില്ല എന്ന യാഥാർത്ഥ്യം പരക്കെ അംഗീകരിക്കപ്പെടുന്നുണ്ട് . കറൻസി ഉപയോഗിക്കുന്നതിലൂടെ ഈ ചെലവുകളും പണമിടപാട് അനിശ്ചിതത്വങ്ങളും വ്യക്തിപരമായി ഒഴിവാക്കാൻ കഴിയും, പക്ഷേ വിശ്വസനീയമായ കക്ഷി ഇല്ലെങ്കിൽ ഒരു ആശയവിനിമയ ചാനലിലൂടെ പണമിടപാടുകൾ നടത്താൻ നിലവിൽ ഒരു സംവിധാനവുമില്ല.

വിശ്വാസത്തിന് പകരം ക്രിപ്റ്റോഗ്രാഫിക് തെളിവിനെ അടിസ്ഥാനമാക്കിയുള്ള ഒരു ഇലക്ട്രോണിക് പണമിടപാട് സംവിധാനമാണ് ഇവിടെ ആവശ്യം, വിശ്വസനീയമായ മൂന്നാം കക്ഷിയുടെ ആവശ്യമില്ലാതെ പരസ്പരം നേരിട്ട് ഇടപാട് നടത്താൻ തൽപരരായ രണ്ട് കക്ഷികളെ ഇത് അനുവദിക്കുന്നു. തിരിച്ചെടുക്കാൻ പ്രായോഗികമായി സാധിക്കാത്ത ഇടപാടുകൾ വിൽപ്പനക്കാരെ കബളിപ്പിക്കലിൽ നിന്ന് സംരക്ഷിക്കും, കൂടാതെ വാങ്ങുന്നവർക്ക് സംരക്ഷണം നൽകുന്നതിന് പതിവ് എസ്ക്രോ സംവിധാനങ്ങൾ എളുപ്പത്തിൽ നടപ്പാക്കാൻ കഴിയും. ഈ പേപ്പറിൽ, ഇടപാടുകളുടെ കാലാനുസൃതമായ ക്രമത്തിന്റെ കണക്കുകളുടെ തെളിവ് ഉണ്ടാക്കുന്നതിന് ഒരു പിയർ-ടു-പിയർ വിതരണ ടൈംസ്റ്റാമ്പ് സെർവർ ഉപയോഗിച്ച് രണ്ടുതവണയുണ്ടാകുന്ന ചെലവ് പ്രശ്നത്തിന് ഒരു പരിഹാരം ഞങ്ങൾ നിർദ്ദേശിക്കുന്നു. ആക്രമണകാരി നോഡുകളുടെ ഏതെങ്കിലും ഗ്രൂപ്പിനേക്കാൾ കൂടുതൽ സിപിയു ശക്തി വിശ്വസ്തമായ നോഡുകൾ ഒരുമിച്ച് നിയന്ത്രിക്കുന്നിടത്തോളം കാലം സിസ്റ്റം സുരക്ഷിതമാണ്.

2 ഇടപാടുകൾ

ഡിജിറ്റൽ ഒപ്പുകളുടെ ഒരു ശൃംഖല എന്നാണ് ഞങ്ങൾ ഒരു ഇലക്ട്രോണിക് നാണയത്തെ നിർവചിക്കുന്നത്. ഓരോ ഉടമയും മുമ്പത്തെ ഇടപാടിന്റെ ഹാഷും അടുത്ത ഉടമയുടെ പബ്ലിക് കീയും ഡിജിറ്റലായി ഒപ്പിട്ട് നാണയത്തിന്റെ അവസാനത്തിലേക്ക് ഇവ ചേർത്തുകൊണ്ട് നാണയം

അടുത്തതിലേക്ക് മാറ്റുന്നു. ഇവിടെ ഉടമസ്ഥാവകാശ ശൃംഖല പരിശോധിക്കുന്നതിന് പണമടയ്ക്കുന്നയാൾക്ക് ഒപ്പുകൾ പരിശോധിക്കാൻ കഴിയും.



ഉടമകളിൽ ഒരാൾ നാണയം രണ്ടുതവണ ചെലവഴിച്ചിട്ടില്ലെന്ന് സ്ഥിരീകരിക്കാൻ പണമടയ്ക്കുന്നയാൾക്ക് കഴിയില്ല എന്നതാണ് നിലവിലെ പ്രശ്നം. ഇരട്ടച്ചെലവിനായി വിശ്വസനീയമായ ഒരു കേന്ദ്ര അതോറിറ്റി അല്ലെങ്കിൽ ഓരോ ഇടപാടും പരിശോധിക്കുന്ന മിന്റ് അവതരിപ്പിക്കുക എന്നതാണ് ഇതിനുള്ള ഒരു പൊതു പരിഹാരം. ഓരോ ഇടപാടിനും ശേഷം, ഒരു പുതിയ നാണയം നൽകുന്നതിന് നാണയം മിന്റിലേക്ക് തിരികെ നൽകണം, എങ്കിലേ മിന്റിൽ നിന്ന് നേരിട്ട് വിതരണം ചെയ്യുന്ന നാണയങ്ങൾ മാത്രമേ രണ്ടുതവണ ചെലവഴിക്കാത്തതായി വിശ്വസിക്കപ്പെടുകയുള്ളൂ. ഈ പരിഹാരത്തിന്റെ ഏക വെല്ലുവിളി മിന്റ് നടത്തുന്ന കമ്പനിയുടെ വിശ്വസ്തതയെ ആശ്രയിച്ചിരിക്കുന്നു, ഓരോ ഇടപാടും ഒരു ബാങ്കിനെപ്പോലെ സുതാര്യമായി നടത്തേണ്ടതുണ്ട്.

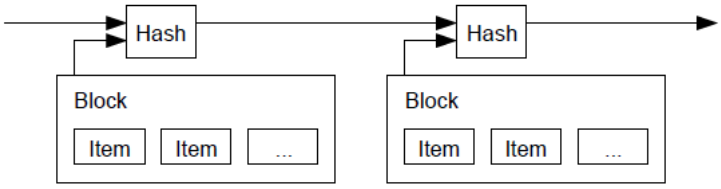
മുൻ ഉടമകൾ മുമ്പത്തെ ഇടപാടുകളിൽ ഒപ്പിട്ടിട്ടില്ലെന്ന് പണമടയ്ക്കുന്നയാൾ അറിയാൻ ഒരു മാർഗം ആവശ്യമാണ്. ഞങ്ങളുടെ ആവശ്യങ്ങൾക്കായി, ഏറ്റവും ആദ്യകാലം മുതലുള്ള ഇടപാടുകൾ കണക്കിലെടുക്കുന്നു, അതിനാൽ രണ്ടുതവണ ചെലവഴിക്കാനുള്ള ശ്രമങ്ങൾ ഞങ്ങൾക്ക് ശ്രദ്ധിക്കേണ്ട ആവശ്യം വരില്ല. ഒരു ഇടപാടിന്റെ അഭാവം സ്ഥിരീകരിക്കാനുള്ള ഒരേയൊരു മാർഗം എല്ലാ ഇടപാടുകളെക്കുറിച്ചും അറിഞ്ഞിരിക്കുക എന്നതാണ്. മിന്റ് അടിസ്ഥാന മാതൃകയിൽ, എല്ലാ ഇടപാടുകളെക്കുറിച്ചും മിന്റ് അറിയുകയും ആദ്യമെത്തിയത് ഏതെന്ന് തീരുമാനിക്കുകയും ചെയ്യുന്നു എന്ന പ്രത്യേകതയുമുണ്ട്. വിശ്വസനീയമായ ഒരു കക്ഷി ഇല്ലാതെ ഇത് നിറവേറ്റുന്നതിന്, ഇടപാടുകൾ പരസ്യമായി പ്രഖ്യാപിക്കണം [1], കൂടാതെ പങ്കാളികൾക്ക് അവരുടെ അംഗീകാര

ക്രമപ്രകാരം ഒരേ സ്ഥലത്ത് രേഖപ്പെടുത്താൻ ഒരു സംവിധാനം ആവശ്യമാണ്. ഓരോ ഇടപാടിന്റെയും സമയത്ത്, ഭൂരിഭാഗം നോഡുകളും ഇത് ആദ്യമായി ലഭിച്ചതാണെന്ന് അംഗീകരിച്ചു എന്നതിന് പണം സ്വീകരിക്കുന്നയാൾക്ക് തെളിവ് ആവശ്യമാണ്.

3 ടൈംസ്റ്റാമ്പ് സെർവർ

ഞങ്ങൾ നിർദ്ദേശിക്കുന്ന പരിഹാരം ഒരു ടൈംസ്റ്റാമ്പ് സെർവറിൽ നിന്ന് ആരംഭിക്കുന്നു. ടൈംസ്റ്റാമ്പ് സെർവർ പ്രവർത്തിക്കുന്നത് ടൈംസ്റ്റാമ്പ് ചെയ്യേണ്ട ഇനങ്ങളുടെ ഒരു വിഭാഗത്തിന്റെ ഹാഷ് എടുത്ത് ഒരു പത്രത്തിലോ യൂസ്നെറ്റ് പോസ്റ്റോ [2-5] പോലുള്ളവയിൽ ഹാഷ് വ്യാപകമായി പ്രസിദ്ധീകരിക്കുന്നതിലൂടെയാണ്. ഹാഷ് ലഭിക്കുന്നന്നതിന് ഡാറ്റ ഉണ്ടായിരുന്നിരിക്കണമെന്ന് ടൈംസ്റ്റാമ്പ് തെളിയിക്കുന്നു. ഓരോ ടൈംസ്റ്റാമ്പിന്റെ ഹാഷിലും അതിന് മുമ്പത്തെ ടൈംസ്റ്റാമ്പ് ഉൾപ്പെടുന്നു,. ഓരോ അധിക ടൈംസ്റ്റാമ്പും അതിന്

മുമ്പുള്ളവയെ വീണ്ടും ശക്തിപ്പെടുത്തിക്കൊണ്ട് ഒരു ശൃംഖല സൃഷ്ടിക്കുന്നു,

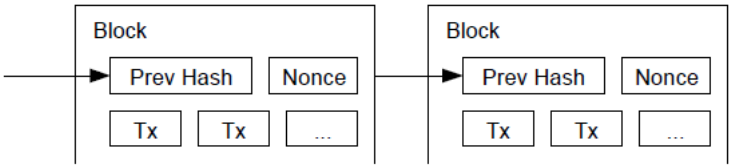


4. പ്രൂഫ് ഓഫ് വർക്ക്

പിയർ-ടു-പിയർ അടിസ്ഥാനത്തിലുള്ള ഒരു ടൈംസ്റ്റാമ്പ് സെർവർ നടപ്പിലാക്കുന്നതിന്, പത്രം അല്ലെങ്കിൽ യൂസ്നെറ്റ് പോസ്റ്റുകൾക്ക് പകരം ആദം ബാക്കിന്റെ ഹാഷ്കാഷിന് [6] സമാനമായ ഒരു പ്രൂഫ്-ഓഫ്-വർക്ക് സിസ്റ്റം ഉപയോഗിക്കേണ്ടതുണ്ട്. SHA-256 പോലുള്ള ഹാഷ് ചെയ്യുമ്പോൾ, ഹാഷ് നിരവധി പൂജ്യം ബിറ്റുകളിൽ നിന്ന് ആരംഭിക്കുന്ന ഒരു മൂല്യത്തിനായി സ്കാൻ ചെയ്യുന്നതിൽ പ്രൂഫ്-ഓഫ്-വർക്ക് ഉൾപ്പെടുന്നു. ആവശ്യമായ പൂജ്യം ബിറ്റുകളുടെ എണ്ണത്തിൽ ആവശ്യമായ ശരാശരി പ്രവർത്തനം എക്സ്പോണൻഷ്യൽ ആണ്, മാത്രമല്ല ഒരൊറ്റ ഹാഷ് നടപ്പിലാക്കുന്നതിലൂടെ ഇത് പരിശോധിക്കാനും കഴിയും..

ഞങ്ങളുടെ ടൈംസ്റ്റാമ്പ് നെറ്റ്വർക്കിനായി, വിഭാഗങ്ങളുടെ ഹാഷിന് ആവശ്യമായ പൂജ്യം ബിറ്റുകൾ നൽകുന്ന ഒരു മൂല്യം കണ്ടെത്തുന്നതുവരെ

വിഭാഗങ്ങളിൽ ഒരു നോൺസെ വർദ്ധിപ്പിച്ചുകൊണ്ട് ഞങ്ങൾ പ്രൂഫ്-ഓഫ്-വർക്ക് നടപ്പിലാക്കുന്നു. പ്രവർത്തനത്തിന് ആവശ്യമായ പൂർണ്ണത ലഭിക്കുന്നതുവരെ സിപിയു പരിശ്രമം നടത്തിക്കഴിഞ്ഞാൽ, പ്രവർത്തനം പുനർനിർമ്മിക്കാതെ വിഭാഗം മാറ്റാനാകില്ല. പിന്നീട് വിഭാഗങ്ങൾ ഇതിനോടൊപ്പം ചേരുന്നതിനാൽ, വിഭാഗം മാറ്റുന്നതിനുള്ള ശ്രമങ്ങളിൽ അതിന് ശേഷമുള്ള എല്ലാ വിഭാഗങ്ങളും പുനർനിർമ്മിക്കുന്നത് ഉൾപ്പെടും.



ഭൂരിപക്ഷ തീരുമാനങ്ങളെടുക്കുന്നതിൽ പ്രാതിനിധ്യം നിർണ്ണയിക്കുന്ന പ്രശ്നവും പ്രൂഫ് ഓഫ് വർക്ക് പരിഹരിക്കുന്നു. ഭൂരിപക്ഷം ഒരു ഐപി-വിലാസം-ഒരു വോട്ടിന്റെ അടിസ്ഥാനത്തിലാണെങ്കിൽ, നിരവധി ഐപികൾ അനുവദിക്കാൻ കഴിയുന്ന ആർക്കും അത് അട്ടിമറിക്കാൻ കഴിയും. പ്രൂഫ് ഓഫ് വർക്ക് അടിസ്ഥാനപരമായി ഒരു-സിപിയു-ഒരു വോട്ട് ആണ്. ഭൂരിപക്ഷ തീരുമാനത്തെ പ്രതിനിധീകരിക്കുന്നത് ഏറ്റവും ദൈർഘ്യമേറിയ ശൃംഖലയാണ്, അതിൽ നിക്ഷേപിച്ച ഏറ്റവും വലിയ പ്രവർത്തന തെളിവ് ഉണ്ട്. സിപിയു ശക്തിയുടെ ഭൂരിഭാഗവും ശരിയായ നോഡുകളാൽ നിയന്ത്രിക്കപ്പെടുകയാണെങ്കിൽ, ശരിയായ ശൃംഖല അതിവേഗം വളരുകയും എല്ലാ മത്സര ശൃംഖലകളെയും മറികടക്കുകയും ചെയ്യും. മുമ്പത്തെ ഒരു വിഭാഗം പരിഷ്കരിക്കുന്നതിന്, ഒരു ആക്രമണകാരിക്ക് എല്ലാ വിഭാഗങ്ങളുടെയും പ്രൂഫ്-ഓഫ്-വർക്ക് പുനർനിർമ്മിക്കുകയും തുടർന്ന് ശരിയായ നോഡുകളുടെ പ്രവർത്തനത്തെ മറികടക്കുകയും വേണം. തുടർന്നുള്ള വിഭാഗങ്ങൾ ചേർക്കുമ്പോൾ ആക്രമണം നടക്കാനുള്ള സാധ്യത ക്രമാതീതമായി കുറയുന്നുവെന്ന് ഞങ്ങൾക്ക് കാണിക്കാനാകും.

ഹാർഡ്വെയറിന്റെ വർദ്ധിച്ചുവരുന്ന വേഗതയും കാലക്രമേണ നോഡുകൾ പ്രവർത്തിപ്പിക്കുന്നതിലെ വ്യത്യസ്ത താൽപ്പര്യവും പരിഹരിക്കുന്നതിന്, മണിക്കൂറിൽ ശരാശരി വിഭാഗങ്ങൾ ലക്ഷ്യമിടുന്ന മാറുന്ന ശരാശരികളാണ് പ്രൂഫ്-ഓഫ്-വർക്കിലെ സങ്കീർണ്ണതയ്ക്ക് കാരണമാകുന്നത്.അവ വേഗത്തിൽ സൃഷ്ടിക്കപ്പെടുകയാണെങ്കിൽ, പ്രതിസന്ധി വർദ്ധിക്കുന്നു.

5. നെറ്റ്വർക്ക്

നെറ്റ്വർക്ക് പ്രവർത്തിപ്പിക്കുന്നതിനുള്ള ഘട്ടങ്ങൾ ഇനിപ്പറയുന്നവയാണ്:

1) പുതിയ ഇടപാടുകൾ എല്ലാ നോഡുകളിലേക്കും പ്രക്ഷേപണം ചെയ്യുന്നു.

2) ഓരോ നോഡും ഒരു വിഭാഗങ്ങളിലേക്ക് പുതിയ ഇടപാടുകൾ ശേഖരിക്കുന്നു.

3) ഓരോ നോഡും അതിന്റെ വിഭാഗത്തിനായി പ്രയാസമുള്ള പ്രൂഫ് ഓഫ് വർക്ക് കണ്ടെത്തുന്നതിന് പ്രവർത്തിക്കുന്നു.

4) ഒരു നോഡ് പ്രൂഫ് ഓഫ് വർക്ക് കണ്ടെത്തുമ്പോൾ, അത് എല്ലാ നോഡുകളിലേക്കും വിഭാഗം പ്രക്ഷേപണം ചെയ്യുന്നു.

5) നോഡുകൾ അതിലെ എല്ലാ ഇടപാടുകളും സാധുതയുള്ളതും ഇതിനകം ചെലവഴിച്ചിട്ടില്ലാത്തതുമാണെങ്കിൽ മാത്രം വിഭാഗം സ്വീകരിക്കുന്നു.

6) അംഗീകൃത വിഭാഗത്തിന്റെ ഹാഷ് മുൻ ഹാഷായി ഉപയോഗിച്ച് ശൃംഖലയിൽ അടുത്ത വിഭാഗം സൃഷ്ടിക്കുന്നതിനായി പ്രവർത്തിച്ചുകൊണ്ട് നോഡുകൾ വിഭാഗത്തിന്റെ സ്വീകാര്യത വ്യക്തമാക്കുന്നു.

നോഡുകൾ എല്ലായ്പ്പോഴും ഏറ്റവും ദൈർഘ്യമേറിയ ശൃംഖല ശരിയായ ഒന്നായി കണക്കാക്കുകയും അത് വിപുലീകരിക്കുന്നതിനായി പ്രവർത്തിക്കുകയും ചെയ്യും. രണ്ട് നോഡുകൾ ഒരേസമയം അടുത്ത വിഭാഗത്തിന്റെ വ്യത്യസ്ത പതിപ്പുകൾ പ്രക്ഷേപണം ചെയ്യുകയാണെങ്കിൽ, ചില നോഡുകൾക്ക് അവയിൽ ഏതെങ്കിലുമൊന്ന് ആദ്യം ലഭിച്ചേക്കാം. അത്തരമൊരു സാഹചര്യത്തിൽ, അവയ്ക്ക് ലഭിച്ച ആദ്യത്തേതിൽ അവ പ്രവർത്തിക്കുന്നു,. എന്നാൽ അതിന്റെ ദൈർഘ്യം കൂടുകയാണെങ്കിൽ മറ്റേ ശാഖ സംരക്ഷിക്കുന്നു. അടുത്ത പ്രൂഫ് ഓഫ് വർക്ക് കണ്ടെത്തുകയും ഒരു ശാഖയ്ക്ക് ദൈർഘ്യം കൂടുകയും ചെയ്യുമ്പോൾ ബന്ധം വിച്ഛേദിക്കപ്പെടുന്നു. മറ്റേ ശാഖയിൽ പ്രവർത്തിച്ചിരുന്ന നോഡുകൾ പിന്നീട് ദൈർഘ്യമേറിയ ഒന്നിലേക്ക് മാറും.

പുതിയ ഇടപാട് പ്രക്ഷേപണങ്ങൾ എല്ലാ നോഡുകളിലും എത്തണമെന്നില്ല. അവ പല നോഡുകളിലും എത്തുന്നിടത്തോളം കാലം, അവ താമസിയാതെ ഒരു വിഭാഗത്തിൽ പ്രവേശിക്കും. വിഭാഗ പ്രക്ഷേപണങ്ങളും ഒഴിവാക്കപ്പെട്ട സന്ദേശങ്ങളെ തളളിക്കളയുന്നില്ല. ഒരു നോഡിന് ഒരു വിഭാഗം ലഭിച്ചില്ലെങ്കിൽ, അടുത്ത വിഭാഗം ലഭിക്കുമ്പോൾ അതിനായി അഭ്യർത്ഥിക്കുകയും അത് നഷ്ടപ്പെട്ടുവെന്ന് മനസ്സിലാക്കുകയും ചെയ്യും.

6. **ഇൻസെന്റീവ്**

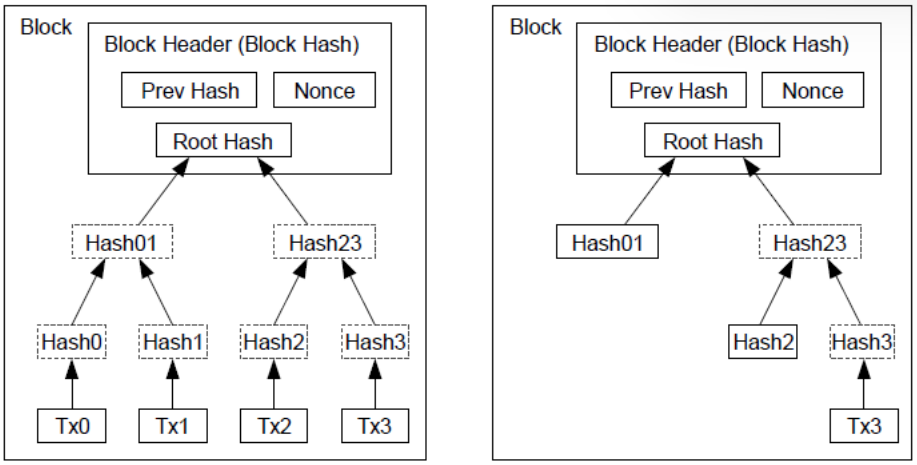
പരമ്പരാഗത രീതി അനുസരിച്ച്, ഒരു വിഭാഗത്തിലെ ആദ്യത്തെ ഇടപാട് വിഭാഗത്തിന്റെ സൃഷ്ടാവിന്റെ ഉടമസ്ഥയിലുള്ള സവിശേഷമായ ഒരു

പുതിയ നാണയമാണ്. ഇത് ശൃംഖലയെ പിന്തുണയ്ക്കുന്നതിന് നോഡുകൾക്ക് ഇൻസന്റീവ് നൽകുന്നു. കൂടാതെ പുറപ്പെടുവിക്കാൻ ഔദ്യോഗിക അധികാരമില്ലാത്തതിൽ തുടക്കത്തിൽ നാണയങ്ങൾ വിതരണം ചെയ്യുന്നതിനുള്ള ഒരു മാർഗം നൽകുന്നു,. പുതിയ നാണയങ്ങളുടെ സ്ഥിരമായ കൂട്ടിച്ചേർക്കൽ സ്വർണ്ണ ഖനിത്തൊഴിലാളികൾ സ്വർണ്ണം ചേർക്കാൻ വിഭവങ്ങൾ ചെലവഴിക്കുന്നതിന് സമാനമാണ്. ഞങ്ങളുടെ കാര്യത്തിൽ, സിപിയു സമയവും വൈദ്യുതിയുമാണ് ചെലവഴിക്കുന്നത്. ഇൻസെന്റീവിന് ഇടപാടിന് പണമടയ്ക്കൽ ഫീസ് ഉപയോഗിച്ചും ധനസഹായം നൽകാം. ഒരു ഇടപാടിന്റെ ഔട്ട്പുട്ട് മൂല്യം അതിന്റെ ഇൻപുട്ട് മൂല്യത്തേക്കാൾ കുറവാണെങ്കിൽ, അതിലെ വ്യത്യാസം, ഇടപാട് അടങ്ങിയ വിഭാഗത്തിന്റെ ഇൻസെന്റീവ് മൂല്യത്തിലേക്ക് ചേർക്കുന്ന ഒരു ഇടപാട് ഫീസാണ്. മുൻകൂട്ടി നിശ്ചയിച്ച എണ്ണം നാണയങ്ങൾ പ്രചാരത്തിലെത്തിയാൽ, ഇൻസെന്റീവ് പൂർണ്ണമായും ഇടപാട് ഫീസിലേക്ക് മാറാനും പൂർണ്ണമായും പണപ്പെരുപ്പ മുക്തമാകാനും കഴിയും.

ശരിയായി തുടരാൻ നോഡുകളെ ഇൻസന്റീവ് സഹായിച്ചേക്കാം. ഒരു ആക്രമണകാരിക്ക് ശരിയായ എല്ലാ നോഡുകളേക്കാളും കൂടുതൽ സിപിയു ശക്തി സമാഹരിക്കാൻ കഴിയുമെങ്കിൽ, തന്റെ പണമിടപാടുകൾ അപഹരിച്ച് ആളുകളെ കബളിപ്പിക്കാൻ ഇത് ഉപയോഗിക്കുന്നതിനോ പുതിയ നാണയങ്ങൾ സൃഷ്ടിക്കാനോ അയാൾക്ക് ശ്രമിക്കാനാകും. വ്യവസ്ഥിതിയെയും സ്വന്തം സമ്പത്തിന്റെ സാധുതയെയും ദുർബലപ്പെടുത്തുന്നതിനേക്കാൾ, മറ്റെല്ലാവരേക്കാളും കൂടുതൽ പുതിയ നാണയങ്ങൾ ഉപയോഗിച്ച് തനിക്ക് അനുകൂലമായ നിയമങ്ങൾ, ഉണ്ടാക്കുന്നത് കൂടുതൽ ലാഭകരമാണെന്ന് തോന്നിയാലാകും അയാൾ ഇതിന് ശ്രമിക്കുക.

7. ഡിസ്ക് സ്ഥലം വീണ്ടെടുക്കൽ

ഒരു നാണയത്തിലെ ഏറ്റവും പുതിയ ഇടപാട് മതിയായ വിഭാഗങ്ങൾക്ക് കീഴിലാക്കിയാൽ, അതിനുമുമ്പ് ചെലവഴിച്ച ഇടപാടുകൾ ഡിസ്ക് സ്ഥലം ലാഭിക്കാനായി നിരാകരിക്കാനാകും. വിഭാഗത്തിന്റെ ഹാഷ് തകർക്കാതെ ഇത് സുഗമമാക്കുന്നതിന്, ഇടപാടുകൾ ഒരു മെർക്കിൾ ട്രീയിൽ [7][2][5] ഹാഷ് ചെയ്യുന്നു, ബ്ലോക്കിന്റെ ഹാഷിൽ റൂട്ടുകൾ മാത്രമേ ഉൾപ്പെടുത്തിയിട്ടുള്ളൂ. വൃക്ഷത്തിന്റെ ശിഖരങ്ങൾ മുറിച്ചുകൊണ്ട് പഴയ വിഭാഗങ്ങൾ പരിമിതപ്പെടുത്താം. ഇതിനായി ഇന്റീരിയർ ഹാഷുകൾ സംഭരിക്കേണ്ട ആവശ്യമില്ല.



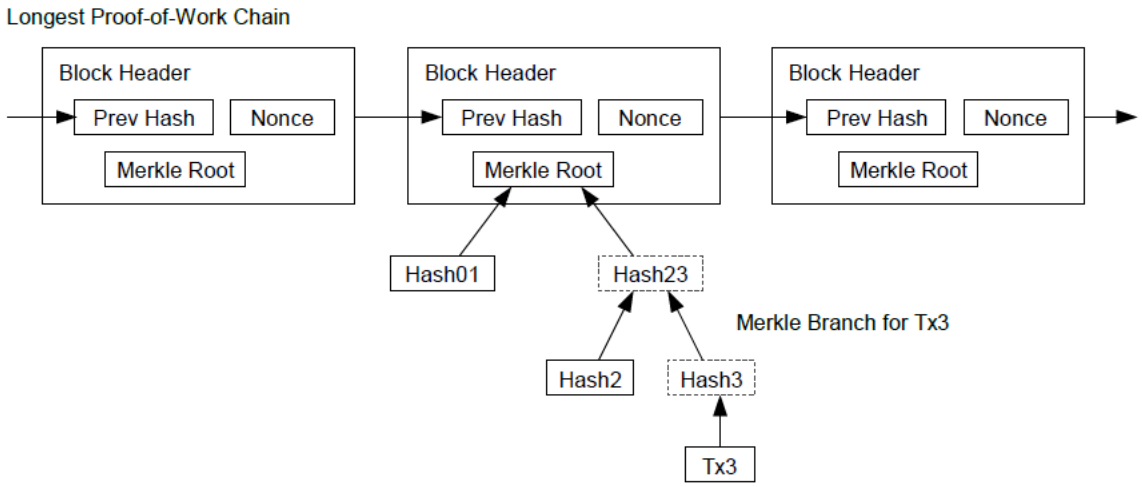
Transactions Hashed in a Merkle Tree

After Pruning Tx0-2 from the Block

ഇടപാടുകളില്ലാത്ത ഒരു വിഭാഗ ഹെഡ്ഡർ ഏകദേശം 80 ബൈറ്റുകൾ ആയിരിക്കും. ഓരോ 10 മിനിറ്റിലും വിഭാഗങ്ങൾ സൃഷ്ടിക്കപ്പെടുന്നുവെന്ന് കരുതുകയാണെങ്കിൽ, 80 ബൈറ്റുകൾ * 6 * 24 * 365 = പ്രതിവർഷം 4.2 MB. 2008 ലെ കണക്കനുസരിച്ച് കമ്പ്യൂട്ടർ സിസ്റ്റങ്ങൾ സാധാരണയായി വിൽപനയ്ക്കായി 2 ജിബി റാം ഉപയോഗിക്കുകയാണെങ്കിൽ, പ്രതിവർഷം 1.2 ജിബിയുടെ നിലവിലെ വളർച്ച പ്രവചിക്കുന്നു. എന്നാൽ മൂർസ് ലോ, വിഭാഗ ഹെഡറുകൾ മെമ്മറിയിൽ സൂക്ഷിക്കേണ്ടതുണ്ടെങ്കിലും സംഭരണം ഒരു പ്രശ്നമാകില്ല.

8. ലളിതമാക്കിയ പണമടയ്ക്കൽ പരിശോധന

ഒരു പൂർണ്ണ നെറ്റ്വർക്ക് നോഡ് പ്രവർത്തിപ്പിക്കാതെ പണമിടപാടുകൾ പരിശോധിക്കാനാകും. ഒരു ഉപയോക്താവ് ഏറ്റവും ദൈർഘ്യമേറിയ പ്രൂഫ്-ഓഫ്-വർക്ക് ശൃംഖലയുടെ വിഭാഗ ഹെഡറുകളുടെ ഒരു പകർപ്പ് മാത്രമേ സൂക്ഷിക്കേണ്ടതുള്ളൂ, അത് തനിക്ക് ഏറ്റവും ദൈർഘ്യമേറിയ ശൃംഖല ഉണ്ടെന്ന് ബോധ്യപ്പെടുന്നതുവരെ നെറ്റ്വർക്ക് നോഡുകൾ അന്വേഷിക്കുന്നതിലൂടെ ലഭിക്കും, കൂടാതെ ഇടപാടിനെ അത് ടൈംസ്റ്റാംപ് ചെയ്ത വിഭാഗവുമായി ബന്ധിപ്പിക്കുന്ന മെർക്കിൾ ബ്രാഞ്ച് നേടുകയും വേണം. അയാൾക്ക് സ്വയം ഇടപാട് പരിശോധിക്കാൻ കഴിയില്ല, പക്ഷേ അത് ശൃംഖലയിലെ ഒരു സ്ഥലത്തേക്ക് ലിങ്കുചെയ്യുന്നതിലൂടെ, ഒരു നെറ്റ്വർക്ക് നോഡ് അത് സ്വീകരിച്ചതായി അയാൾക്ക് കാണാൻ കഴിയും, കൂടാതെ നെറ്റ്വർക്ക് അത് സ്വീകരിച്ചുവെന്ന് കൂടുതൽ സ്ഥിരീകരിക്കുന്ന വിഭാഗങ്ങളും ഉണ്ടാകും.

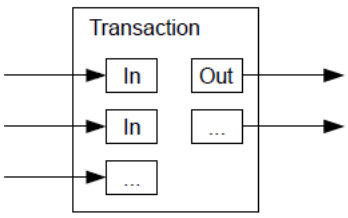


ഏറ്റവും ദൈർഘ്യമേറിയ പ്രൂഫ്-ഓഫ്-വർക്ക് ശൃംഖല

Tx3-ന് വേണ്ടിയുള്ള മെർക്കിൾ ബ്രാഞ്ച്

അതിനാൽ, ശരിയായ നോഡുകൾ നെറ്റ്‌വർക്കിനെ നിയന്ത്രിക്കുന്നിടത്തോളം കാലം പരിശോധന വിശ്വസനീയമാണ്, പക്ഷേ നെറ്റ്‌വർക്ക് ഒരു ആക്രമണകാരി കീഴടക്കിയാൽ കൂടുതൽ ദുർബലമായേക്കാം. നെറ്റ്‌വർക്ക് നോഡുകൾക്ക് സ്വയം ഇടപാടുകൾ പരിശോധിക്കാൻ കഴിയുമെങ്കിലും, ആക്രമണകാരിക്ക് നെറ്റ്‌വർക്കിനെ മറികടക്കുന്നത് തുടരാൻ കഴിയുന്നിടത്തോളം കാലം ആക്രമണകാരിക്ക് വ്യാജ ഇടപാടുകളാൽ കബളിപ്പിക്കാൻ കഴിയും. അസാധുവായ ഒരു വിഭാഗം കണ്ടെത്തുമ്പോൾ നെറ്റ്‌വർക്ക് നോഡുകളിൽ നിന്നുള്ള അലേർട്ടുകൾ സ്വീകരിക്കുക എന്നതാണ് ഇതിൽ നിന്ന് പരിരക്ഷ നേടുന്നതിനുള്ള ഒരു തന്ത്രം, ഇത് ഉപയോക്താവിന്റെ സോഫ്റ്റ്‌വെയറിനെ മുഴുവൻ വിഭാഗവും ഡൗൺലോഡ് ചെയ്യാനും പൊരുത്തക്കേട് സ്ഥിരീകരിക്കുന്നതിന് അലേർട്ട് ചെയ്ത ഇടപാടുകൾ നടത്താൻ പ്രേരിപ്പിക്കുന്നു. പതിവായി പണമിടപാടുകൾ നടത്തുന്ന ബിസിനസുകൾ കൂടുതൽ സ്വതന്ത്രമായ സുരക്ഷയ്ക്കും വേഗത്തിലുള്ള പരിശോധനയ്ക്കുമായി സ്വന്തം നോഡുകൾ പ്രവർത്തിപ്പിക്കാൻ ശ്രമിച്ചേക്കാം.

9. സംയോജനവും വിഭജന മൂല്യവും

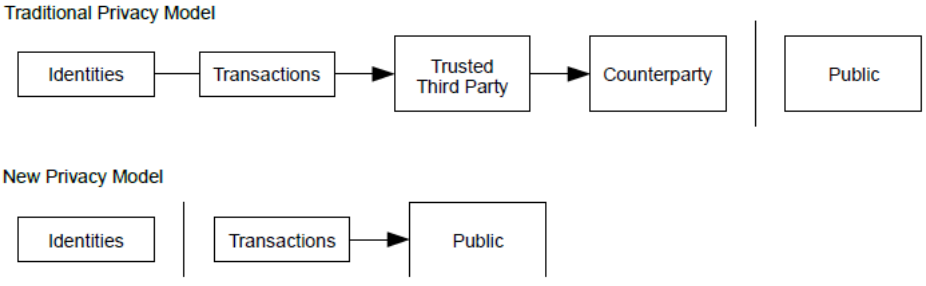


നാണയങ്ങൾ വ്യക്തിഗതമായി കൈകാര്യം ചെയ്യാൻ കഴിയുമെങ്കിലും, കൈമാറ്റത്തിലെ ഓരോ സെന്റിനും പ്രത്യേക ഇടപാട് നടത്തുന്നത് അസാധ്യമായിരിക്കും. മൂല്യം വിഭജിക്കാനും സംയോജിപ്പിക്കാനും അനുവദിക്കുന്നതിന്, ഇടപാടുകളിൽ ഒന്നിലധികം ഇൻപുട്ടുകളും ഔട്ട്പുട്ടുകളും അടങ്ങിയിരിക്കുന്നു. സാധാരണഗതിയിൽ ഒരു വലിയ മുൻ ഇടപാടിൽ നിന്ന് ഒരൊറ്റ ഇൻപുട്ട് അല്ലെങ്കിൽ ചെറിയ തുകകൾ സംയോജിപ്പിക്കുന്ന ഒന്നിലധികം ഇൻപുട്ടുകൾ ഉണ്ടാകും, പരമാവധി രണ്ട് ഔട്ട്പുട്ടുകൾ ഉണ്ടാകും: ഒന്ന് പണമിടപാടിനായി, മറ്റൊന്ന് എന്തെങ്കിലും മാറ്റം ഉണ്ടായാൽ അയയ്ക്കുന്നയാൾക്ക് തിരികെ നൽകുന്നു.

ഒരു ഇടപാട് മറ്റ് നിരവധി ഇടപാടുകളെ ആശ്രയിച്ചിരിക്കുന്നു. ആ ഇടപാടുകൾ കൂടുതൽ കാര്യങ്ങളെ ആശ്രയിക്കുന്നതുമായ ഫാൻ-ഔട്ട് ഇവിടെ വിഷയമല്ല എന്നത് ശ്രദ്ധിക്കേണ്ടതാണ്. ഒരു ഇടപാടിന്റെ ചരിത്രത്തിന്റെ പൂർണ്ണമായ പകർപ്പ് വേർതിരിച്ചെടുക്കേണ്ട ആവശ്യമില്ല

10. സ്വകാര്യത

പരമ്പരാഗത ബാങ്കിംഗ് മാതൃക ഉൾപ്പെട്ട കക്ഷികൾക്കും വിശ്വസനീയമായ മൂന്നാം കക്ഷിക്കും വിവരങ്ങളിലേക്കുള്ള പ്രവേശനം പരിമിതപ്പെടുത്തുന്നതിലൂടെ സ്വകാര്യതയുടെ ഒരു തലം കൈവരിക്കുന്നു. എല്ലാ ഇടപാടുകളും പരസ്യമായി പ്രഖ്യാപിക്കേണ്ടതിന്റെ ആവശ്യകത ഇവിടെ ബാധകമല്ല, പക്ഷേ മറ്റൊരു സ്ഥലത്തെ വിവരങ്ങളുടെ ഒഴുക്ക് ഇല്ലാതാക്കുന്നതിലൂടെ സ്വകാര്യത നിലനിർത്താൻ കഴിയും: പൊതു കീകൾ രഹസ്യമായി സൂക്ഷിക്കുന്നതിലൂടെ. ആരെങ്കിലും മറ്റൊരാൾക്ക് ഒരു തുക അയയ്ക്കുന്നുവെന്ന് പൊതുജനങ്ങൾക്ക് കാണാൻ കഴിയും, പക്ഷേ ഇടപാടിനെ മറ്റാരുമായും ബന്ധിപ്പിക്കുന്ന വിവരങ്ങൾ ഇല്ലാതെ. ഇത് സ്റ്റോക്ക് എക്സ്ചേഞ്ചുകൾ പുറത്തുവിട്ട വിവരങ്ങളുടെ നിലവാരത്തിന് സമാനമാണ്, അവിടെ വ്യക്തിഗത ട്രേഡുകളുടെ സമയവും വലുപ്പവും, "ടേപ്പ്" പരസ്യപ്പെടുത്തുന്നു, പക്ഷേ കക്ഷികൾ ആരാണെന്ന് പറയാതെയാണെന്ന് മാത്രം



പരമ്പരാഗത സ്വകാര്യത മാതൃക

പുതിയ സ്വകാര്യതാ മാതൃക

ഒരു അധിക സുരക്ഷ എന്ന നിലയിൽ, ഒരു സാധാരണ ഉടമയുമായി ബന്ധിപ്പിക്കുന്നത് തടയാൻ ഓരോ ഇടപാടിനും ഒരു പുതിയ കീ ജോഡി ഉപയോഗിക്കണം. മൾട്ടി-ഇൻപുട്ട് ഇടപാടുകളുമായി ചില ലിങ്കിംഗ് ഇപ്പോഴും ഒഴിവാക്കാനാവില്ല, ഇത് അവരുടെ ഇൻപുട്ടുകൾ ഒരേ ഉടമയുടെ ഉടമസ്ഥതയിലാണെന്ന് വെളിപ്പെടുത്തുന്നു. ഒരു ഉടമയെ വെളിപ്പെടുത്തിയാൽ, ലിങ്കുചെയ്യുന്നത് അതേ ഉടമയുടെ മറ്റ് ഇടപാടുകൾ വെളിപ്പെടുത്താനാകും എന്നതാണ് ഏക പോരായ്മ.

11. **കണക്കുകൂട്ടലുകൾ**

ശരിയായ ശൃംഖലയേക്കാൾ വേഗത്തിൽ ഒരു ബദൽ ശൃംഖല സൃഷ്ടിക്കാൻ ശ്രമിക്കുന്ന ഒരു ആക്രമണകാരിയുടെ സാഹചര്യം ഞങ്ങൾ പരിഗണിക്കുന്നു. ഇത് നടന്നാലും, ആക്രമണകാരിക്ക് ഒരിക്കലും സ്വന്തമല്ലാത്ത പണം എടുക്കുക തുടങ്ങിയ ഏകപക്ഷീയമായ സാഹചര്യങ്ങൾകക് ഇത് സിസ്റ്റത്തെ അനുവദിക്കുന്നില്ല. നോഡുകൾ ഒരു അസാധുവായ ഇടപാട് പേയ്മെന്റായി സ്വീകരിക്കാൻ പോകുന്നില്ല, ശരിയായ നോഡുകൾ അവ അടങ്ങിയ ഒരു ബ്ലോക്ക് ഒരിക്കലും സ്വീകരിക്കില്ല. ഒരു ആക്രമണകാരിക്ക് അടുത്തിടെ ചെലവഴിച്ച പണം തിരിച്ചെടുക്കാൻ സ്വന്തം ഇടപാടുകളിലൊന്ന് മാറ്റാൻ ശ്രമിക്കാനേ കഴിയൂ.

ശരിയായ ശൃംഖലയും ആക്രമണ ശൃംഖലയും തമ്മിലുള്ള മത്സരത്തെ ഒരു ബൈനോമിയൽ റാൻഡം വാക്ക് എന്ന് വിശേഷിപ്പിക്കാം. ശരിയായ ശൃംഖല ഒരു ബ്ലോക്കിലേക്ക് നീട്ടുകയും അതിന്റെ ലീഡ് +1 വർദ്ധിപ്പിക്കുകയും, പരാജയം ആക്രമണകാരിയുടെ ശൃംഖല ഒരു വിഭാഗത്തിലേക്ക് നീട്ടുകയും വിടവ് -1 കുറയ്ക്കുകയും ചെയ്യുന്നു.

ഒരു നിശ്ചിത കുറവിൽ നിന്ന് ഒരു ആക്രമണകാരി പിടികൂടാനുള്ള സാധ്യത ഒരു ചൂതാട്ടക്കാരന്റെ നാശത്തിന് സമാനമാണ്. പരിധിയില്ലാത്ത വായ്പയുള്ള ഒരു ചൂതാട്ടക്കാരൻ കുറവിൽ നിന്ന് ആരംഭിക്കുകയും ബ്രേക്ക് ഈവനിൽ എത്താൻ ശ്രമിക്കുന്നതിന് അനന്തമായ പരീക്ഷണങ്ങൾ നടത്തുകയും ചെയ്യുന്നുവെന്ന് കരുതുക. അയാൾ എപ്പോഴെങ്കിലും ബ്രേക്ക് ഈവനിൽ എത്താനുള്ള സാധ്യത നമുക്ക് കണക്കാക്കാം, അല്ലെങ്കിൽ ഒരു ആക്രമണകാരി ശരിയായ ശൃംഖലയിൽ, ഇനിപ്പറയുന്ന രീതിയിൽ എപ്പോഴെങ്കിലും എത്തുന്നു.

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

p > q എന്ന ഞങ്ങളുടെ അനുമാനം കണക്കിലെടുക്കുമ്പോൾ, ആക്രമണകാരിക്ക് കയറിക്കൂടേണ്ട വിഭാഗങ്ങളുടെ എണ്ണം വർദ്ധിക്കുന്നതിനനുസരിച്ച് സാധ്യത ക്രമാതീതമായി കുറയുന്നു. അയാൾക്കെതിരായ പ്രതിബന്ധങ്ങൾ കണക്കിലെടുക്കുമ്പോൾ, തുടക്കത്തിൽ തന്നെ ഭാഗ്യം ലഭിച്ചില്ലെങ്കിൽ, അയാൾ കൂടുതൽ പിന്നിലാകുമ്പോൾ അയാളുടെ സാധ്യതകൾ അപ്രത്യക്ഷമാകും.

ഒരു പുതിയ ഇടപാടിന്റെ സ്വീകർത്താവ് അയച്ചയാൾക്ക് ഇടപാട് മാറ്റാൻ കഴിയില്ലെന്ന് വേണ്ടത്ര ഉറപ്പ് ലഭിക്കുന്നതിന് എത്ര സമയം കാത്തിരിക്കേണ്ടതുണ്ടെന്ന് ഞങ്ങൾ ഇപ്പോൾ പരിഗണിക്കുന്നു. അയയ്ക്കുന്നയാൾ ഒരു ആക്രമണകാരിയാണെന്ന് ഞങ്ങൾ അനുമാനിക്കുന്നു, സ്വീകർത്താവിന് കുറച്ച് സമയത്തേക്ക് പണം നൽകിയെന്ന് വിശ്വസിപ്പിക്കാൻ ആഗ്രഹിക്കുന്നു, തുടർന്ന് കുറച്ച് സമയം കഴിഞ്ഞതിന് ശേഷം സ്വയം തിരിച്ചടയ്ക്കാൻ അത് മാറ്റുക. അത് സംഭവിക്കുമ്പോൾ സ്വീകർത്താവിന് മുന്നറിയിപ്പ് നൽകും, പക്ഷേ അത് വളരെ വൈകുമെന്ന് അയയ്ക്കുന്നയാൾ പ്രതീക്ഷിക്കുന്നു.

സ്വീകർത്താവ് ഒരു പുതിയ കീ ജോഡി സൃഷ്ടിക്കുകയും ഒപ്പിടുന്നതിന് തൊട്ടുമുമ്പ് അയച്ചയാൾക്ക് പബ്ലിക് കീ നൽകുകയും ചെയ്യുന്നു. വേണ്ടത്ര മുന്നോട്ട് പോകാൻ ഭാഗ്യം ലഭിക്കുന്നതുവരെ തുടർച്ചയായി പ്രവർത്തിച്ചുകൊണ്ട് സമയത്തിന് മുമ്പായി വിഭാഗങ്ങളുടെ ഒരു ശൃംഖല തയ്യാറാക്കുന്നതിൽ നിന്ന് അയയ്ക്കുന്നയാളെ ഇത് തടയുന്നു, തുടർന്ന് ആ നിമിഷത്തിൽ ഇടപാട് നടപ്പിലാക്കുന്നു. ഇടപാട് അയച്ചുകഴിഞ്ഞാൽ, വ്യാജമായി അയയ്ക്കുന്നയാൾ തന്റെ ഇടപാടിന്റെ ഇതര പതിപ്പ് അടങ്ങിയ ഒരു സമാന്തര ശൃംഖലയിൽ രഹസ്യമായി പ്രവർത്തിക്കാൻ തുടങ്ങുന്നു.

ഇടപാട് ഒരു വിഭാഗത്തിലേക്ക് ചേർക്കുകയും അതിനുശേഷം ഇസഡ് ബ്ലോക്കുകൾ ലിങ്കുചെയ്യുകയും ചെയ്യുന്നതുവരെ സ്വീകർത്താവ് കാത്തിരിക്കും. ആക്രമണകാരി കൈവരിച്ച പുരോഗതിയുടെ കൃത്യമായ അയാൾക്ക് അറിയില്ല, പക്ഷേ ശരിയായ വിഭാഗങ്ങൾ കണക്കാക്കാൻ ശരാശരി പ്രതീക്ഷിച്ച സമയം എടുത്തു

$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Solving for P less than 0.1%...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. ഉപസംഹാരം

ഇലക്ട്രോണിക് ഇടപാടുകൾക്കായി വിശ്വാസത്തെ ആശ്രയിക്കാത്ത ഒരു സംവിധാനം ഞങ്ങൾ മുന്നോട്ട് വയ്ക്കുന്നു. ഡിജിറ്റൽ ഒപ്പുകളിൽ നിന്ന് നിർമ്മിച്ച നാണയങ്ങളുടെ സാധാരണ ചട്ടക്കൂടിൽ നിന്നാണ് ഞങ്ങൾ ഇത് ആരംഭിച്ചത്. ഇത് ഉടമസ്ഥാവകാശത്തിനു ശക്തമായ നിയന്ത്രണം നൽകുന്നു, പക്ഷേ രണ്ടുതവണയുള്ള ചെലവ് തടയുന്നതിനുള്ള മാർഗമില്ലാതെ അപൂർണ്ണമാണ്. ഇത് പരിഹരിക്കുന്നതിന്, ഇടപാടുകളുടെ ഒരു പൊതു ചരിത്രം രേഖപ്പെടുത്തുന്നതിന് പ്രൂഫ്-ഓഫ്-വർക്ക് ഉപയോഗിച്ച് ഒരു പിയർ-ടു-പിയർ നെറ്റ്വർക്ക് ഞങ്ങൾ നിർദ്ദേശിച്ചു, ശരിയായ നോഡുകൾ സിപിയു ശക്തിയുടെ ഭൂരിഭാഗവും നിയന്ത്രിക്കുകയാണെങ്കിൽ ആക്രമണകാരിക്ക് പെട്ടെന്ന് ഇടപെടുന്നത് അസാധ്യമാക്കുന്നു. ഘടനാരഹിതമായി സങ്കീർണ്ണതകൾ കുറവായതിനാൽ ശൃംഖല ശക്തമാണ്. നോഡുകൾ ചെറിയ ഏകോപനത്തോടെ ഒരേസമയം പ്രവർത്തിക്കുന്നു.

സന്ദേശങ്ങൾ ഏതെങ്കിലും പ്രത്യേക സ്ഥലത്തേക്ക് റൂട്ട് ചെയ്യാത്തതിനാൽ അവ തിരിച്ചറിയേണ്ട ആവശ്യമില്ല. നോഡുകൾക്ക് ഇഷ്ടാനുസരണം നെറ്റ്‌വർക്കിൽ ചേരാനും കഴിയും, അവ പുറത്തു പോയപ്പോൾ എന്താണ് സംഭവിച്ചതെന്നതിന്റെ തെളിവായി പ്രൂഫ്-ഓഫ്-വർക്ക് ശൃംഖല സ്വീകരിക്കുന്നു. അവർ തങ്ങളുടെ സിപിയു അധികാരം ഉപയോഗിച്ച് വോട്ടുചെയ്യുന്നു, സാധുവായ ബ്ലോക്കുകൾ വിപുലീകരിക്കുന്നതിൽ പ്രവർത്തിക്കുന്നതിലൂടെ തങ്ങളുടെ സ്വീകാര്യത വ്യക്തമാക്കുകയും അവയിൽ പ്രവർത്തിക്കാൻ വിസമ്മതിച്ച് അസാധുവായ വിഭാഗങ്ങൾ നിരാകരിക്കുകയും ചെയ്യുന്നു. ആവശ്യമായ നിയമങ്ങളും പ്രോത്സാഹനങ്ങളും ഈ സമവായ സംവിധാനം ഉപയോഗിച്ച് നടപ്പാക്കാൻ കഴിയും.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.