ବିଟ୍‌କଏନ: ଏକ ପିୟର-ଟୁ-ପିୟର ଇଲେକ୍‌ଟ୍ରୋନିକ କ୍ୟାଶ ସିଷ୍ଟମ

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

ଆବ୍‌ଷ୍ଟ୍ରାକ୍ଟ: ଇଲେକ୍‌ଟ୍ରୋନିକ୍ କ୍ୟାଶର ଏକ ପିୟର-ଟୁ-ପିୟର (ପି୨ପି) ଭର୍ସନ ଏକ ଆର୍ଥିକ ଅନୁଷ୍ଠାନର ସହଯୋଗ ବିନା ଗୋଟିଏ ପକ୍ଷରୁ ଅନ୍ୟ ଏକ ପକ୍ଷକୁ ସିଧାସଳଖ ଅନ୍‌ଲାଇନ ପେମେଂଟ ପଠାଇବା ପାଇଁ ଅନୁମତି ଦେବ। ଡିଜିଟାଲ ସିଗ୍ନେଚରଗୁଡିକ ସମାଧାନର ଏକ ଅଂଶ ପ୍ରଦାନ କରିଥାନ୍ତି କିନ୍ତୁ ଯଦି ଏବେ ବି ଦୁଇ ଗୁଣ ବ୍ୟୟକୁ ରୋକିବା ପାଇଁ ଏକ ବିଶ୍ୱସନୀୟ ୩ୟ ପକ୍ଷର ଆବଶ୍ୟକ ପଡୁଛି ତେବେ ମୁଖ୍ୟ ଲାଭ ମିଳିପାରିବ ନାହିଁ। ଆମେ ଏକ ପିୟର-ଟୁ-ପିୟର ନେଟ୍‌ୱର୍କର ବ୍ୟବହାର କରି ଡବଲ-ସ୍ପେଣ୍ଡିଙ୍ଗ ବା ଦୁଇ ଗୁଣ ବ୍ୟୟ ସମସ୍ୟା ପାଇଁ ଏକ ସମାଧାନର ପ୍ରସ୍ତାବ ଦେଇଛୁ। ଏହି ନେଟ୍‌ୱର୍କ ହାଶ୍-ଆଧାରିତ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କର ଏକ ବିଦ୍ୟାମାନ ଶୃଙ୍ଖଳାରେ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ହାଶ୍ କରି ସେଗୁଡିକୁ ଟାଇମ୍‌ଷ୍ଟାମ୍ପ କରିଥାଏ, ଏକ ରେକର୍ଡ ଗଠନ କରିଥାଏ ଯାହାକୁ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କକୁ ପୁଣିଥରେ ନକରି ପରିବର୍ତନ କରାଯାଇପାରିବ ନାହିଁ। ସବୁଠାରୁ ଦୀର୍ଘ ଶୃଙ୍ଖଳା କେବଳ ପରିଲକ୍ଷିତ ହୋଇଥିବା ଇଭେଂଟ୍‌ଗୁଡିକର କ୍ରମର ପ୍ରମାଣ ଭାବେ କାର୍ଯ୍ୟ କରେ ନାହିଁ ବରଂ ଏହା ସିପିୟୁ ପାୱାରର ସର୍ବବୃହତ୍ ନେଟ୍‌ୱର୍କରୁ ଆସିଛି ବୋଲି ପ୍ରମାଣ ଦେଇଥାଏ। ଯେପର୍ଯ୍ୟନ୍ତ ଅଧିକାଂଶ ସିପିୟୁ ପାୱାର, ନେଟ୍‌ୱର୍କକୁ ଆକ୍ରମଣ କରିବା ପାଇଁ ସହଯୋଗ କରୁନଥିବା ନୋଡ୍‌ଗୁଡିକ ଦ୍ୱାରା ନିୟନ୍ତ୍ରିତ ହେଉଛନ୍ତି, ସେଗୁଡିକ ସବୁଠାରୁ ଦୀର୍ଘ ଶୃଙ୍ଖଳା ସୃଷ୍ଟି କରିବେ ଓ ଆକ୍ରମଣକାରୀଙ୍କୁ ପଛରେ ପକାଇବେ। ଏହି ନେଟ୍‌ୱର୍କ ନିଜେ ସର୍ବନିମ୍ନ ଢାଂଚାର ଆବଶ୍ୟକ କରିଥାଏ। ଏଥିରେ ମେସେଜଗୁଡିକ ସର୍ବୋତମ ପ୍ରୟାସ ଆଧାରରେ ପ୍ରସାରିତ ହୋଇଥାଏ ଏବଂ ନିଜ ଇଚ୍ଛାରେ ନେଟ୍‌ୱର୍କ ଛାଡି ପାରୁଥିବା ଓ ପୁନଃ ଯୋଗ ଦେଇପାରୁଥିବା ନୋଡ୍‌ଗୁଡିକ ସେମାନେ ଯିବା ପରେ କ'ଣ ଘଟିଥିଲା ତାହାର ପ୍ରମାଣ ଭାବେ ଦୀର୍ଘତମ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ଶୃଙ୍ଖଳାକୁ ଗ୍ରହଣ କରିପାରନ୍ତି।

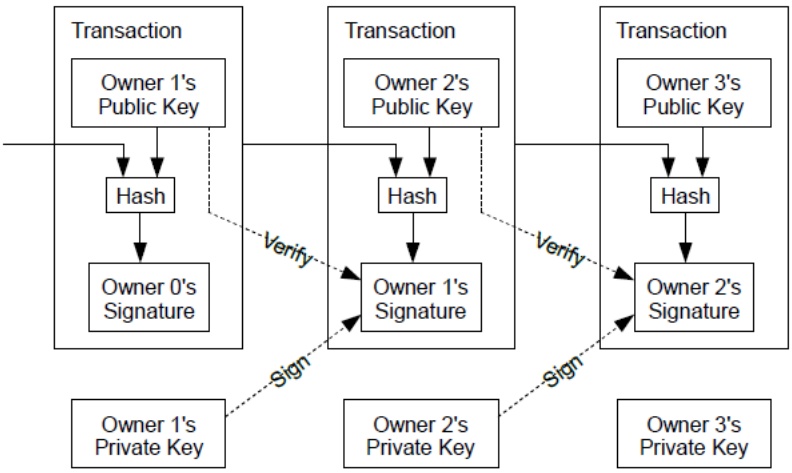
୧. ପରିଚୟ

ଇଂଟରନେଟ୍‌ରେ ବ୍ୟବସାୟ, ଇଲେକ୍‌ଟ୍ରୋନିକ୍ସ ପେମେଂଟ୍‌ଗୁଡିକୁ ପ୍ରକ୍ରିୟାକରଣ କରିବା ପାଇଁ ବିଶ୍ୱସନୀୟ ୩ୟ ପକ୍ଷ ଭାବେ କାର୍ଯ୍ୟ କରୁଥିବା ଆର୍ଥିକ ଅନୁଷ୍ଠାନଗୁଡିକ ଉପରେ ସ୍ୱତନ୍ତ୍ର ଭାବେ ନିର୍ଭରଶୀଳ ଅଟନ୍ତି। ଅଧିକାଂଶ ଟ୍ରାଞ୍ଜାକ୍ସନ ପାଇଁ ଏହି ସିଷ୍ଟମ ଯଥେଷ୍ଟ ଭଲ କାମ କରୁଥିବା ବେଳେ ଏହା ଏବେ ବି ବିଶ୍ୱାସ ଆଧାରିତ ମଡେଲଗୁଡିକର ଅନ୍ତର୍ନିହିତ ଦୁର୍ବଳତା ଦ୍ୱାରା ପ୍ରଭାବିତ ହେଉଛି। ସମ୍ପୂର୍ଣ୍ଣ ରୂପେ ନନ୍-ରିଭର୍ସିବୁଲ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ବାସ୍ତବରେ ସମ୍ଭବ ନୁହେଁ, କାରଣ ଆର୍ଥିକ ଅନୁଷ୍ଠାନଗୁଡିକ ବିବାଦରେ ମଧ୍ୟସ୍ଥତାକୁ ଏଡାଇ ପାରିବେ ନାହିଁ। ମଧ୍ୟସ୍ଥତାର ଖର୍ଚ୍ଚ ଟ୍ରାଞ୍ଜାକ୍ସନ ଖର୍ଚ୍ଚକୁ ବୃଦ୍ଧି କରିଥାଏ, ସର୍ବନିମ୍ନ ବ୍ୟବହାରିକ ଟ୍ରାଞ୍ଜାକ୍ସନ ଆକାରକୁ ସୀମିତ କରିଥାଏ ଏବଂ କ୍ଷୁଦ୍ର ସାଧାରଣ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ପାଇଁ ସମ୍ଭାବନାକୁ ହ୍ରାସ କରିଥାଏ, ଏବଂ ନନ୍-ରିଭର୍ସିବୁଲ ସର୍ଭିସଗୁଡିକ ପାଇଁ ନନ୍-ରିଭର୍ସିବୁଲ ପେମେଂଟ୍ କରିବାର କ୍ଷମତା ହରାଇବା ସ୍ଥିତିରେ ଅଧିକ ଅର୍ଥ ବ୍ୟୟ କରିବାକୁ ପଡିଥାଏ। ରିଭର୍ସାଲର ସମ୍ଭାବନା ସହିତ, ବିଶ୍ୱାସ ପାଇଁ ଆବଶ୍ୟକତା ବ୍ୟାପକ ହୋଇଥାଏ। ବ୍ୟବସାୟୀମାନେ ନିଜର ଗ୍ରାହକଙ୍କଠାରୁ ସତର୍କ ରହିବା ଉଚିତ୍, ଯିଏକି ତାଙ୍କୁ ଆବଶ୍ୟକତାଠାରୁ ଅଧିକ ସୂଚନା ପାଇଁ ହଇରାଣ କରିଥାନ୍ତି। ଠକାମିର ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ହାରକୁ ଅନିବାର୍ଯ୍ୟ ଭାବେ ଗ୍ରହଣ କରାଯାଏ। ଏହି ଖର୍ଚ୍ଚ ଓ ପେମେଂଟ ସମ୍ଭାବନାକୁ ଫିଜିକାଲ କରେନ୍ସିର ବ୍ୟବହାର କରି ବ୍ୟକ୍ତିଗତ ଭାବେ ଏଡା ଯାଇପାରିବ, କିନ୍ତୁ କୌଣସି ବିଶ୍ୱସନୀୟ ପକ୍ଷ ବିନା ଏକ କମ୍ୟୁନିକେଶନ ଚ୍ୟାନେଲ ମାଧ୍ୟମରେ ପେମେଂଟ୍ କରିବା ପାଇଁ କୌଣସି ମେକାନିଜିମ ବା ପ୍ରଣାଳୀ ଉପଲବ୍ଧ ନାହିଁ।

ବିଶ୍ୱାସ ପରିବର୍ତ୍ତେ କ୍ରିପ୍ଟୋଗ୍ରାଫିକ୍ ପ୍ରମାଣ ଉପରେ ଆଧାରିତ ଏକ ଇଲେକ୍ଟ୍ରୋନିକ ପେମେଂଟ ସିଷ୍ଟମର ଆବଶ୍ୟକତା ରହିଛି, ଯାହା ଯେକୌଣସି ଦୁଇଟି ସହମତ ପକ୍ଷକୁ ଏକ ବିଶ୍ୱସନୀୟ ୩ୟ ପକ୍ଷର ଆବଶ୍ୟକତା ବିନା ପରସ୍ପର ସହ ସିଧାସଳଖ ଟ୍ରାଞ୍ଜାକ୍ସନ କରିବା ପାଇଁ ଅନୁମତି ଦେଉଥିବ। ରିଭର୍ସ କରିବା ପାଇଁ ପରିସଂଖ୍ୟା ଦୃଷ୍ଟିରୁ ଅଯୌକ୍ତିକ ମନେ ହେଉଥିବା ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ସେଲରଙ୍କୁ ଠକେଇରୁ ସୁରକ୍ଷା ପ୍ରଦାନ କରିବେ ଏବଂ ନିୟମିତ ଏସ୍କ୍ରୋ ମେକାନିଜିମ୍‌କୁ କ୍ରେତାଙ୍କୁ ସୁରକ୍ଷିତ ରଖିବା ପାଇଁ ସହଜରେ କାର୍ଯ୍ୟକାରୀ କରାଯାଇପାରିବ। ଏହି ପେପର୍‌ରେ ଆମେ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକର କ୍ରୋନୋଲୋଜିକାଲ ଅର୍ଡରର କମ୍ପ୍ୟୁଟେଶନାଲ ପ୍ରୁଫ୍ ସୃଷ୍ଟି କରିବା ପାଇଁ ଏକ ପିୟର-ଟୁ-ପିୟର ଡିଷ୍ଟ୍ରିବ୍ୟୁଟେଡ ଟାଇମ୍‌ଷ୍ଟାମ୍ପର ବ୍ୟବହାର କରି ଦୁଇ ଗୁଣ ଖର୍ଚ୍ଚ ସମସ୍ୟା ପାଇଁ ଏକ ସମାଧାନର ପ୍ରସ୍ତାବ ଉପସ୍ଥାପନ କରିଛୁ। ଯେପର୍ଯ୍ୟନ୍ତ ପ୍ରକୃତ ନୋଡ୍‌ଗୁଡିକ, ଆକ୍ରମଣକାରୀ ନୋଡ୍‌ଗୁଡିକର ଯେକୌଣସି କୋଅପରେଟିଂ ଗ୍ରୁପ୍ ତୁଳନାରେ ସାମୂହିକ ଭାବେ ଅଧିକ ସିପିୟୁ ପାୱାରକୁ ନିୟନ୍ତ୍ରଣ କରିବେ ସେପର୍ଯ୍ୟନ୍ତ ଏହି ସିଷ୍ଟମ ସୁରକ୍ଷିତ ଅଟେ।

୨. ଟ୍ରାଞ୍ଜାକ୍ସନ

ଆମେ ଏକ ଇଲେକ୍ଟ୍ରୋନିକ କଏନକୁ ଡିଜିଟାଲ ସିଗ୍ନେଚରଗୁଡିକର ଏକ ଶୃଙ୍ଖଳା ଭାବେ ବ୍ୟାଖ୍ୟା କରୁଛୁ। ପ୍ରତ୍ୟେକ ମାଲିକ ପୂର୍ବ ଟ୍ରାଞ୍ଜାକ୍ସନର ଏକ ହାଶ୍ ଓ ପରବର୍ତୀ ମାଲିକଙ୍କ ସାର୍ବଜନିନ ଚାବିକୁ ଡିଜିଟାଲ ଉପାୟରେ ସାଇନ କରି ପରବର୍ତୀ ଟ୍ରାଞ୍ଜାକ୍ସନ ପାଇଁ କଏନକୁ ଟ୍ରାନ୍ସଫର କରିଥାନ୍ତି ଏବଂ ଏଗୁଡିକୁ କଏନ୍‌ର ଶେଷରେ ଯୋଗ କରିଥାନ୍ତି। ପେଇ ବା ଅର୍ଥ ପାଇଥିବା ବ୍ୟକ୍ତି ମାଲିକାନାର ଶୃଙ୍ଖଳାକୁ ଯାଂଚ କରିବା ପାଇଁ ସିଗ୍ନେଚରଗୁଡିକୁ ଯାଂଚ କରିପାରିବେ।



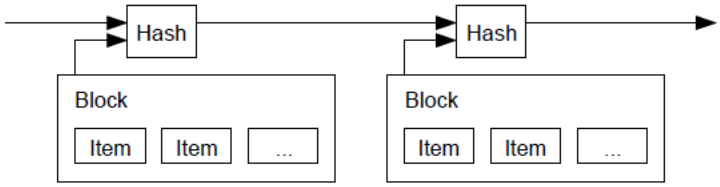
ତେବେ ସମସ୍ୟା ହେଉଛି ପେଇ ବା ଅର୍ଥ ପାଇଥିବା ବ୍ୟକ୍ତି କୌଣସି ମାଲିକ କଏନକୁ ଦୁଇ ଗୁଣ ଖର୍ଚ୍ଚ କରିଛନ୍ତି କି ନାହିଁ ତାହାକୁ ଯାଂଚ କରିପାରିବେ ନାହିଁ। ଏହାର ଏକ ସାଧାରଣ ସମସ୍ୟା ହେଉଛି ଏକ ବିଶ୍ୱସନୀୟ ସେଂଟ୍ରାଲ ଅଥୋରିଟି କିମ୍ୱା ମିଂଟ୍‌ର ଆରମ୍ଭ କରିବା, ଯାହା ଦୁଇ ଗୁଣ ଖର୍ଚ୍ଚ ସମ୍ପର୍କରେ ଜାଣିବା ପାଇଁ ପ୍ରତ୍ୟେକ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଯାଂଚ କରିବ। ପ୍ରତ୍ୟେକ ଟ୍ରାଞ୍ଜାକ୍ସନ ପରେ କଏନକୁ ନିଶ୍ଚିତ ରୂପେ ଏକ ନୂଆ କଏନ ଜାରି କରିବା ପାଇଁ ମିଂଟ୍‌କୁ ଫେରସ୍ତ କରାଯିବା ଉଚିତ ଏବଂ ସିଧାସଳଖ ମିଂଟରୁ ଜାରି କରାଯାଇଥିବା କଏନଗୁଡିକୁ ହିଁ ଦୁଇ ଗୁଣ ଖର୍ଚ୍ଚ କରାଯାଇନାହିଁ ବୋଲି ବିଶ୍ୱାସ କରାଯିବ। ଏହି ସମାଧାନ ସହ ଜଡିତ ସମସ୍ୟା ହେଉଛି ସମଗ୍ର ଅର୍ଥ ପ୍ରଣାଳୀର ଭାଗ୍ୟ ମିଂଟ୍‌କୁ ପରିଚାଳନା କରୁଥିବା କମ୍ପାନୀ ଉପରେ ନିର୍ଭର କରିଥାଏ, କାରଣ ପ୍ରତ୍ୟେକ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଠିକ୍ ଏକ ବ୍ୟାଙ୍କ ଭଳି ଉକ୍ତ କମ୍ପାନୀ ମାଧ୍ୟମରେ ପ୍ରୋସେସ ହେବାକୁ ପଡିଥାଏ।

ଅର୍ଥ ପାଉଥିବା ବ୍ୟକ୍ତିଙ୍କ ଲାଗି ପୂର୍ବ ମାଲିକମାନେ କୌଣସି ପୂର୍ବ ଟ୍ରାଞ୍ଜାକ୍ସନ ସାଇନ କରିନାହାନ୍ତି ବୋଲି ଜାଣିବା ପାଇଁ ଆମେ ଏକ ମାଧ୍ୟମର ଏକ ଆବଶ୍ୟକ କରୁଛୁ। ଆମର ଉଦ୍ଦେଶ୍ୟ ପାଇଁ ପ୍ରାରମ୍ଭିକ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ହିଁ ଗଣନା କରାଯାଇଥାଏ, ଯାହାଫଳରେ ଆମେ ଡବଲ-ସ୍ପେଣ୍ଡ ବା ଦୁଇଗୁଣ ଖର୍ଚ୍ଚ ପାଇଁ ପରବର୍ତୀ ସମୟରେ କରାଯାଇଥିବା ପ୍ରୟାସଗୁଡିକୁ ନେଇ ଚିନ୍ତିତ ହେବୁ ନାହିଁ। ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନର ଅନୁପସ୍ଥିତିକୁ ନିଶ୍ଚିତ କରିବା ପାଇଁ ଏକମାତ୍ର ଉପାୟ ହେଉଛି ସମସ୍ତ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ସମ୍ପର୍କରେ ସତର୍କ ରହିବା। ମିଂଟ୍ ଆଧାରିତ ମଡେଲରେ ମିଂଟ୍ ସମସ୍ତ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ନେଇ ସତର୍କ ରହିଥାଏ ଏବଂ କେଉଁ ଟ୍ରାଞ୍ଜାକ୍ସନ ପ୍ରଥମେ ଆସିଛି ତାହାର ନିଷ୍ପତି ନେଇଥାଏ। ଏକ ବିଶ୍ୱସନୀୟ ପକ୍ଷ ବିନା ଏହାକୁ ହାସଲ କରିବା ପାଇଁ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ନିଶ୍ଚିତ ରୂପେ ସାର୍ବଜନିନ ଭାବେ ଘୋଷଣା କରାଯିବା ଉଚିତ ଏବଂ ଆମେ ଅର୍ଡର୍‌ର ଏକ ସିଙ୍ଗଲ୍ ହିଷ୍ଟ୍ରି ଉପରେ ସହମତ ହେବା ପାଇଁ ଅଂଶଗ୍ରହଣକାରୀଙ୍କ ଲାଗି ଏକ ସିଷ୍ଟମର ଆବଶ୍ୟକ କରୁଛୁ। ପେଇ

ବା ଟଙ୍କା ପାଉଥିବା ବ୍ୟକ୍ତି ପ୍ରମାଣ ଆବଶ୍ୟକ କରନ୍ତି ଯେ, ପ୍ରତ୍ୟେକ ଟ୍ରାଞ୍ଜାକ୍ସନ ସମୟରେ ଅଧିକାଂଶ ନୋଡ୍ ଏହା ଗ୍ରହଣ କରାଯାଇଥିବା ପ୍ରଥମ ଟ୍ରାଞ୍ଜାକ୍ସନ ବୋଲି ସହମତ ହୋଇଥିଲେ।

୩. ଟାଇମ୍‌ଷ୍ଟାମ୍ପ ସର୍ଭର

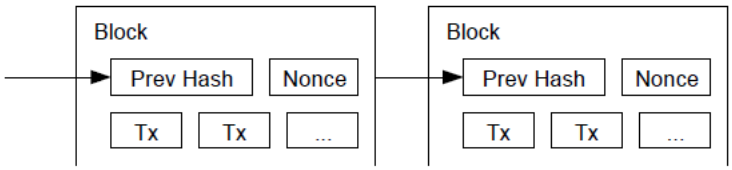
ଆମେ ପ୍ରସ୍ତାବିତ କରିଥିବା ସମାଧାନ ଏକ ଟାଇମ୍‌ଷ୍ଟାମ୍ପ ସର୍ଭର ସହ ଆରମ୍ଭ ହେଉଛି। ଏକ ଟାଇମଷ୍ଟାମ୍ପ ସର୍ଭର, ଟାଇମଷ୍ଟାମ୍ପ ହେବାକୁ ଥିବା ଆଇଟମଗୁଡିକର ଏକ ବ୍ଲକର ହାଶ୍‌କୁ ନେଇ କାର୍ଯ୍ୟ କରିଥାଏ ଏବଂ ଏକ ଖବରକାଗଜ କିମ୍ବା ୟୁଜ୍‌ନେଟ୍ ପୋଷ୍ଟ ଭଳି ବ୍ୟାପକ ରୂପେ ହାଶକୁ ପବ୍ଲିଶ କରିଥାଏ। ଟାଇମ୍‌ଷ୍ଟାମ୍ପ ପ୍ରମାଣିତ କରିଥାଏ ଯେ, ହାଶକୁ ଯିବା ପାଇଁ ଡାଟା ସେହି ସମୟରେ ନିଶ୍ଚିତ ରୂପେ ବିଦ୍ୟମାନ ରହିଥିବା ଉଚିତ। ପ୍ରତ୍ୟେକ ଟାଇମଷ୍ଟାମ୍ପ ନିଜ ହାଶ୍‌ରେ ପୂର୍ବର ଟାଇମ୍‌ଷ୍ଟାମ୍ପକୁ ଅନ୍ତର୍ଭୁକ୍ତ କରିଥାଏ, ଯାହା ଏକ ଚେନ୍ ବା ଶୃଙ୍ଖଳା ଗଠନ କରିଥାଏ, ପ୍ରତ୍ୟେକ ଅତିରିକ୍ତ ଟାଇମଷ୍ଟାମ୍ପ ଏହା ପୂର୍ବର ଟାଇମ୍‌ଷ୍ଟାମ୍ପକୁ ସୁଦୃଢ କରିଥାଏ।



୪. ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ

ଏକ ବିତରଣ ହୋଇଥିବା ଟାଇମ୍‌ଷ୍ଟାମ୍ପ ସର୍ଭରକୁ ଏକ ପିୟର-ଟୁ-ପିୟର ଆଧାରରେ କାର୍ଯ୍ୟକାରୀ କରିବା ପାଇଁ ଆମକୁ ଖବରକାଗଜ କିମ୍ବା ୟୁଜ୍‌ନେଟ ପୋଷ୍ଟ ପରିବର୍ତେ ଆଡାମ ବ୍ୟାକ୍‌ଙ୍କ ହାଶ୍‌କ୍ୟାଶ ଭଳି ସମାନ ଏକ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ସିଷ୍ଟମର ବ୍ୟବହାର କରିବାକୁ ପଡିବ। ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କରେ ଏକ ମୂଲ୍ୟ ପାଇଁ ସ୍କାନିଙ୍ଗ ଅନ୍ତର୍ଭୁକ୍ତ ରହିଥାଏ , ଯାହାକି ଯେତେବେଳେ ହାଶ କରାଯାଇଥାଏ, ଏସ୍‌ଏଚ୍‌ଏ-୨୫୬ ଭଳି, ଉକ୍ତ ହାଶ୍ ଜିରୋ ବିଟ୍‌ସର ଏକ ନମ୍ବର ସହ ଆରମ୍ଭ ହୋଇଥାଏ । ଆବଶ୍ୟକ ହାରାହାରି କାର୍ଯ୍ୟ ଜିରୋ ବିଟ୍‌ସର ସଂଖ୍ୟାରେ ବୃଦ୍ଧି ପାଇଥାଏ ଏବଂ ଏହାକୁ ଏକ ସିଙ୍ଗଲ ହାଶ୍‌କୁ କାର୍ଯ୍ୟକାରୀ କରି ଯାଂଚ କରାଯାଇପାରିବ।

ଆମର ଟାଇମ୍‌ଷ୍ଟାମ୍ପ ନେଟ୍‌ୱର୍କ ପାଇଁ ଆମେ ଏକ ମୂଲ୍ୟ ନ ମିଳିବା ପର୍ଯ୍ୟନ୍ତ ବ୍ଲକ୍‌ରେ ଏକ ନୋନ୍ସକୁ କାର୍ଯ୍ୟକାରୀ କରି ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କକୁ କାର୍ଯ୍ୟକାରୀ କରିଥାଉ ଯାହା ବ୍ଲକ୍‌ର ହାଶ୍‌କୁ ଆବଶ୍ୟକୀୟ ଜିରୋ ବିଟ୍‌ସ ଦେଇଥାଏ। ଥରେ ସିପିୟୁର ପ୍ରୟାସ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କକୁ ସନ୍ତୋଷଜନକ କରିବା ପାଇଁ ବ୍ୟୟ ହୋଇଗଲେ, କାର୍ଯ୍ୟକୁ ପୁଣିଥରେ ନକରି ବ୍ଲକ୍‌କୁ ପରିବର୍ତନ କରାଯାଇପାରିବ ନାହିଁ। ଯେହେତୁ ପରବର୍ତୀ ବ୍ଲକ୍‌ଗୁଡିକ ଏହା ପରେ ଶୃଙ୍ଖଳାରେ ସାମିଲ ହୋଇଥାନ୍ତି, ତେଣୁ ବ୍ଲକ୍‌କୁ ପରିବର୍ତନ କରିବା କାର୍ଯ୍ୟରେ ସମସ୍ତ ବ୍ଲକ୍‌ଗୁଡିକୁ ଏହାପରେ ରିଡୁଙ୍ଗ୍ କରିବା ସାମିଲ ରହିବ।



ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ଅଧିକାଂଶ ନିଷ୍ପତି ଗ୍ରହଣକାରୀ କାର୍ଯ୍ୟରେ ପ୍ରତିନିଧିତ୍ୱ ନିର୍ଦ୍ଧାରଣ କରିବା ସମସ୍ୟାର ସମାଧାନ କରିଥାଏ। ଯଦି ସଂଖ୍ୟାଗରିଷ୍ଠତା ୱାନ-ଆଇପି-ଆଡ୍ରେସ୍-ୱାନ-ଭୋଟ ଉପରେ ଆଧାରିତ ଥାଏ, ତେବେ ଏହାକୁ ଅନେକ ଆଇପିକୁ ଆବଂଟିତ କରିବା ପାଇଁ ସକ୍ଷମ ଯେକୌଣସି ବ୍ୟକ୍ତିଙ୍କ ଦ୍ୱାରା ନଷ୍ଟ କରାଯାଇପାରିବ। ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ହେଉଛି ଗୁରୁତ୍ୱପୂର୍ଣ୍ଣ ଭାବେ ୱାନ୍-ସିପିୟୁ-ୱାନ୍-ଭୋଟ୍। ଅଧିକାଂଶ ନିଷ୍ପତି ଦୀର୍ଘତମ ଶୃଙ୍ଖଳା ଦ୍ୱାରା ପ୍ରତିନିଧିତ୍ୱ ହୋଇଥାଏ, ଯେଉଁଥିରେ ଏଥିରେ ବିନିଯୋଗ ହୋଇଥିବା ସର୍ବବୃହତ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ପ୍ରୟାସ ରହିଛି। ଯଦି ସିପିୟୁ ପାୱାରର ଅଧିକାଂଶକୁ ପ୍ରକୃତ ନୋଡ୍‌ଗୁଡିକ ଦ୍ୱାରା ନିୟନ୍ତ୍ରଣ କରାଯାଏ ତେବେ ସଚ୍ଚୋଟ ଶୃଙ୍ଖଳା ଦ୍ରୁତ ବେଗରେ ବୃଦ୍ଧି ପାଇବ ଏବଂ ଯେକୌଣସି ପ୍ରତିଯୋଗୀ ଶୃଙ୍ଖଳାକୁ ପଛରେ ପକାଇବ। ଏକ ପୂର୍ବ ବ୍ଲକକୁ ରୂପାନ୍ତରଣ କରିବା ପାଇଁ ଜଣେ ଆକ୍ରମଣକାରୀଙ୍କୁ ବ୍ଲକ୍ ଓ ଏହାପରବର୍ତୀ ସମସ୍ତ ବ୍ଲକ୍‌କୁ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କକୁ ରିଡୁ ବା ପୁଣିଥରେ କରିବାକୁ ପଡିବ ଏବଂ ଏହାପରେ ସଚ୍ଚୋଟ ନୋଡ୍‌ଗୁଡିକର କାର୍ଯ୍ୟ ସହ ମେଳ ଖାଇବା ଓ ତାହାକୁ ଅତିକ୍ରମ କରିବାକୁ ପଡିବ। ଆମେ ପରବର୍ତୀ ସମୟରେ ପ୍ରଦର୍ଶିତ କରିବୁ ଯେ, ପରବର୍ତୀ ବ୍ଲକ୍‌ଗୁଡିକୁ ଯୋଗ କରାଯାଉଥିବାରୁ, ଜଣେ ମନ୍ଥର ଆକ୍ରମଣକାରୀଙ୍କ ସଚ୍ଚୋଟ ନୋଟ୍‌ଗୁଡିକର କାର୍ଯ୍ୟ ନିକଟରେ ପହଂଚିବାର ସମ୍ଭାବନା ବହୁମାତ୍ରାରେ ହ୍ରାସ ପାଇଥାଏ।

ସମୟକ୍ରମେ ନୋଟ୍‌ଗୁଡିକୁ ପରିଚାଳନା କରିବା କ୍ଷେତ୍ରରେ ବର୍ଦ୍ଧିତ ହାର୍ଡୱେୟାର ସ୍ପିଡ୍ ଓ ବିଭିନ୍ନ ଆଗ୍ରହ ପାଇଁ କ୍ଷତିପୂରଣ ଦେବା ଲାଗି ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କର ଜଟିତଳାକୁ ପ୍ରତି ଘଂଟା ପିଛା ହାରାହାରି ବ୍ଲକ୍ ସଂଖ୍ୟାକୁ ଲକ୍ଷିତ କରୁଥିବା ଏକ ଗତିଶୀଳ ହାରାହାରି ସଂଖ୍ୟା ଦ୍ୱାରା ନିର୍ଦ୍ଧାରଣ କରାଯାଇଥାଏ।

୫. ନେଟ୍‌ୱର୍କ

ନେଟ୍‌ୱର୍କକୁ ପରିଚାଳନା କରିବା ପାଇଁ ପଦକ୍ଷେପଗୁଡିକ ନିମ୍ନରେ ଉଲ୍ଲେଖ ରହିଛି:

୧) ନୂଆ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ସମସ୍ତ ନୋଡ୍ ପାଇଁ ପ୍ରସାରଣ କରାଯାଇଥାଏ।

୨) ପ୍ରତ୍ୟେକ ନୋଡ୍ ନୂଆ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ଏକ ବ୍ଲକ୍‌ରେ ସଂଗ୍ରହ କରିଥାଏ।

୩) ପ୍ରତ୍ୟେକ ନୋଡ୍ ଏହାର ବ୍ଲକ୍ ପାଇଁ ଏକ ଜଟିଳ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ସନ୍ଧାନ କରିବା ଉପରେ କାର୍ଯ୍ୟ କରିଥାଏ।

୪) ଯେତେବେଳେ ନୋଡ୍ ଏକ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କର ସନ୍ଧାନ କରିଥାଏ, ଏହା ସମସ୍ତ ନୋଡ୍ ପାଇଁ ବ୍ଲକ୍‌କୁ ପ୍ରସାରଣ କରିଥାଏ।

୫) ଯଦି ଏଥିରେ ଥିବା ସମସ୍ତ ଟ୍ରାଞ୍ଜାକ୍ସନ ବୈଧ ଅଟେ ଏବଂ ବ୍ୟୟ ହୋଇନାହିଁ ତେବେ ନୋଡ୍ ବ୍ଲକ୍‌କୁ ଗ୍ରହଣ କରିଥାଏ।

୬) ନୋଡ୍‌ଗୁଡିକ ଶୃଙ୍ଖଳାରେ ପରବର୍ତୀ ବ୍ଲକ୍ ସୃଷ୍ଟି କରିବା ଉପରେ କାର୍ଯ୍ୟ କରି, ଗ୍ରହଣ କରିଥିବା ବ୍ଲକ୍‌ର ହାଶ୍‌କୁ ପୂର୍ବ ହାଶ୍ ଭାବେ ବ୍ୟବହାର କରି, ବ୍ଲକ୍‌କୁ ଗ୍ରହଣ କରିଥାନ୍ତି।

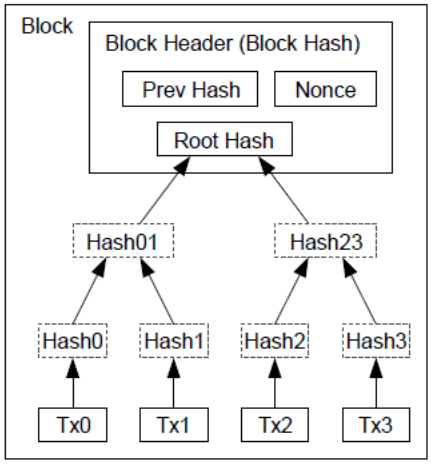
ନୋଡ୍‌ଗୁଡିକ ସବୁବେଳେ ଦୀର୍ଘତମ ଶୃଙ୍ଖଳାକୁ ସଠିକ୍ ଶୃଙ୍ଖଳା ଭାବେ ବିଚାର କରିଥାନ୍ତି ଏବଂ ଏହାକୁ ବିସ୍ତାର କରିବା ଉପରେ କାର୍ଯ୍ୟ କରିଥାନ୍ତି। ଯଦି ଦୁଇଟି ନୋଡ୍ ଏକାସଙ୍ଗେ ପରବର୍ତୀ ବ୍ଲକ୍‌ର ଭିନ୍ନ ଭିନ୍ନ ଭର୍ସନକୁ ପ୍ରସାରଣ କରନ୍ତି ତେବେ କିଛି ନୋଡ୍ ଗୋଟିଏ କିମ୍ବା ଅନ୍ୟଟିକୁ ପ୍ରଥମ ବ୍ଲକ୍ ଭାବେ ଗ୍ରହଣ କରିପାରନ୍ତି। ସେଭଳି କ୍ଷେତ୍ରରେ ସେମାନେ ଗ୍ରହଣ କରିଥିବା ପ୍ରଥମ ବ୍ଲକ୍ ଉପରେ କାର୍ଯ୍ୟ କରିଥାନ୍ତି, କିନ୍ତୁ ଯଦି ଏହା ଅଧିକ ଲମ୍ବା ହୋଇଯାଏ ତେବେ ଅନ୍ୟ ବ୍ରାଂଚ୍‌କୁ ସଂରକ୍ଷଣ କରି ରଖିଥାନ୍ତି। ପରବର୍ତୀ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ମିଳିବା ପରେ ଓ ଗୋଟିଏ ବ୍ରାଂଚ୍ ଅଧିକ ଲମ୍ବା ହୋଇଗଲେ, ଟାଇ ବା ବନ୍ଧନ ଭାଙ୍ଗିଯାଇଥାଏ; ଏହାପରେ ଅନ୍ୟ ବ୍ରାଂଚ୍ ଉପରେ କାର୍ଯ୍ୟ କରୁଥିବା ନୋଡ୍‌ଗୁଡିକ ଅଧିକ ଲମ୍ବା ଶାଖାକୁ ସ୍ଥାନାନ୍ତରିତ ହୋଇଥାନ୍ତି।

ନୂଆ ଟ୍ରାଞ୍ଜାକ୍ସନର ପ୍ରସାରଣ ସମସ୍ତ ନୋଡ୍ ନିକଟରେ ପହଂଚିବା ଜରୁରୀ ନୁହେଁ। ଯେପର୍ଯ୍ୟନ୍ତ ସେମାନେ ଅନେକ ନୋଡରେ ନପହଂଚିଛନ୍ତି, ସେମାନେ ବହୁପୂର୍ବରୁ ଏକ ବ୍ଲକ୍‌ରେ ପ୍ରବେଶ କରିବେ। ବ୍ଲକ୍ ପ୍ରସାରଣଗୁଡିକ ଡ୍ରପ୍ ହୋଇଥିବା ମେସେଜଗୁଡିକ ପ୍ରତି ସହନଶୀଳ ଅଟେ। ଯଦି ଏକ ନୋଡ୍ ଏକ ବ୍ଲକ୍ ଗ୍ରହଣ ନକରେ ତେବେ ଏହା ପରବର୍ତୀ ବ୍ଲକ୍ ଗ୍ରହଣ କରିବା ସମୟରେ ଏହାକୁ ଅନୁରୋଧ କରିବ ଏବଂ ଏହା ଗୋଟିଏ ବ୍ଲକ୍ ମିସ୍ କରିଛି ବୋଲି ଅନୁଭବ କରିବ।

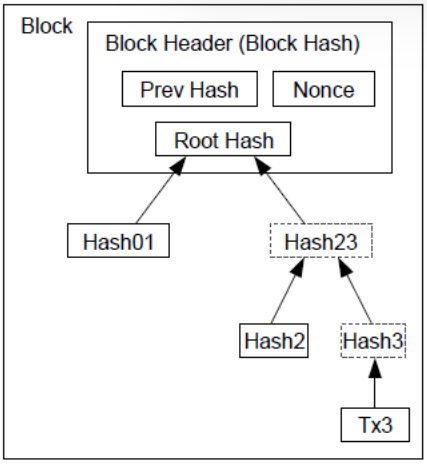
୬. ଇନ୍‌ସେଂଟିଭ

ପରମ୍ପରା ଅନୁଯାୟୀ, ଏକ ବ୍ଲକ୍‌ରେ ପ୍ରଥମ ଟ୍ରାଞ୍ଜାକ୍ସନ ହେଉଛି ଏକ ସ୍ୱତନ୍ତ୍ର ଟ୍ରାଞ୍ଜାକ୍ସନ ଯାହା ଏକ ନୂଆ କଏନ ଆରମ୍ଭ କରିଥାଏ ଯାହା ବ୍ଲକ୍‌ର ନିର୍ମାତାଙ୍କ ମାଲିକାନାରେ ଥାଏ। ଏହା ନେଟ୍‌ୱର୍କକୁ ସମର୍ଥନ କରିବା ପାଇଁ ନୋଡ୍‌ଗୁଡିକ ଲାଗି ଏକ ପ୍ରୋତ୍ସାହନକୁ ଯୋଗ କରିଥାଏ ଏବଂ ପ୍ରାରମ୍ଭିକ ରୂପେ କଏନଗୁଡିକୁ ବିତରଣ କରିବା ପାଇଁ ଏକ ମାଧ୍ୟମ ପ୍ରଦାନ କରିଥାଏ, ଯେହେତୁ ସେଗୁଡିକୁ ଜାରି କରିବା ପାଇଁ କୌଣସି ସେଂଟ୍ରାଲ ଅଥୋରିଟି ନାହିଁ। ନୂଆ କଏନଗୁଡିକର ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ପରିମାଣର ସ୍ଥିର ଯୋଗ, ସୁନା ଖଣି ମାଲିକଙ୍କ ସହ ସମାନ ହୋଇଥାଏ, ଯିଏକି ବିତରଣରେ ଅଧିକ ସୁନା ଯୋଗ କରିବା ପାଇଁ ସଂଶାଧନଗୁଡିକୁ ବ୍ୟୟ କରିଥାନ୍ତି। ଆମ କ୍ଷେତ୍ରରେ, ଏହା ହେଉଛି ସିପିୟୁ ଟାଇମ ଓ ଇଲେକ୍ଟ୍ରିସିଟି ଯାହା ବ୍ୟୟ ହୋଇଥାଏ।

ଇନ୍‌ସେଂଟିଭ୍‌କୁ ଟ୍ରାଜାକ୍ସନ ଫିସ୍ ସହିତ ମଧ୍ୟ ପ୍ରଦାନ କରାଯାଇପାରିବ। ଯଦି ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନର ଆଉଟ୍‌ପୁଟ୍ ଭାଲ୍ୟୁ ଏହାର ଇନ୍‌ପୁଟ ଭାଲ୍ୟୁଠାରୁ କମ୍ ରହିଛି ତେବେ ଡିଫରେନ୍ସ ବା ପାର୍ଥକ୍ୟ ହେଉଛି ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନ ଫି' ଯାହାକୁ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଧାରଣ କରିଥିବା ବ୍ଲକ୍‌ର ଇନ୍‌ସେଂଟିଭ୍ ମୂଲ୍ୟରେ ଯୋଗ କରାଯାଇଥାଏ। ଥରେ କଏନଗୁଡିକର ଏକ ପୂର୍ବ ନିର୍ଦ୍ଧାରିତ ସଂଖ୍ୟା ସର୍କୁଲେଶନରେ ପ୍ରବେଶ କରିବା ପରେ, ଇନ୍‌ସେଂଟିଭ୍ ସମ୍ପୂର୍ଣ୍ଣ ରୂପେ ଟ୍ରାଞ୍ଜାକ୍ସନ ଫି'କୁ ଯାଇପାରେ ଏବଂ ସମ୍ପୂର୍ଣ୍ଣ ରୂପେ ମୁଦ୍ରାସ୍ଫୀତି ମୁକ୍ତ ହୋଇପାରେ।



Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

ଇନ୍‌ସେଂଟିଭ୍ ନୋଡ୍‌ଗୁଡିକୁ ସଚ୍ଚୋଟ ରହିବା ପାଇଁ ପ୍ରୋତ୍ସାହିତ କରିବାରେ ସାହାଯ୍ୟ କରିପାରନ୍ତି। ଯଦି ଜଣେ ଲୋଭୀ ଆକ୍ରମଣକାରୀ ସମସ୍ତ ସଚ୍ଚୋଟ ନୋଡ୍ ତୁଳନାରେ ଅଧିକ ସିପିୟୁ ପାୱାର ଏକତ୍ରିତ କରିବାକୁ ସକ୍ଷମ ହୁଅନ୍ତି, ତେବେ ତାଙ୍କୁ ଏହାକୁ ନିଜର ପେମେଂଟ୍‌ଗୁଡିକୁ ପୁଣିଥରେ ଚୋରି କରି ଲୋକମାନଙ୍କୁ ଠକିବା ପାଇଁ ବ୍ୟବହାର କରିବା କିମ୍ବା ଏହାକୁ ନୂଆ କଏନ ସୃଷ୍ଟି କରିବା ପାଇଁ ବ୍ୟବହାର କରିବା ମଧ୍ୟରେ ଚୟନ କରିବାକୁ ପଡିବ। ସେ ନିୟମର ଅନୁପାଳନ କରି ଏହାକୁ ଅଧିକ ଲାଭଦାୟକ ବିଚାର କରିବା ଉଚିତ, ଏଭଳି ନିୟମ ଯାହା ସିଷ୍ଟମ ଓ ନିଜ ସମ୍ପତିର ବୈଧତାକୁ କ୍ଷୁର୍ଣ୍ଣ କରିବା ଅପେକ୍ଷା ସମସ୍ତଙ୍କ ଏକତ୍ରିତ କଏନ ତୁଳନାରେ ଅଧିକ ସଂଖ୍ୟକ ନୂତନ କଏନ ସହ ତାଙ୍କୁ ଆକର୍ଷିତ କରୁଥିବ।

୭. ରିକ୍ଲେମିଙ୍ଗ ଡିସ୍କ ସ୍ପେଶ୍

ଥରେ ଏକ କଏନରେ ସଦ୍ୟତମ ଟ୍ରାଞ୍ଜାକ୍ସନ ପର୍ଯ୍ୟାପ୍ତ ପରିମାଣର ବ୍ଲକ୍ ତଳେ ଦବି ହୋଇଗଲା ପରେ, ଏହାପୂର୍ବରୁ ବ୍ୟୟ ହୋଇଥିବା ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ଡିସ୍କର ସ୍ପେଶକୁ ସଂଚୟ କରିବା ପାଇଁ ବର୍ଜନ କରାଯାଇପାରିବ। ବ୍ଲକ୍ର ହାଶ୍କୁ ନଭାଙ୍ଗି ଏହାକୁ ସୁଗମ କରିବା ପାଇଁ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ଏକ ମର୍କେଲ ଟ୍ରୀରେ ହାଶ୍ କରାଯାଇଥାଏ, କେବଳ ରୁଟ୍କୁ ବ୍ଲକ୍ର ହାଶ୍ରେ ଅନ୍ତର୍ଭୁକ୍ତ କରାଯାଇଥାଏ। ପରେ ପୁରୁଣା ବ୍ଲକଗୁଡିକୁ ଟ୍ରୀର ବ୍ରାଂଚ୍ଗୁଡିକୁ କାଟି ସୁଦୃଢ କରାଯାଇପାରିବ। ଇଂଟେରିୟର ହାଶ୍ଗୁଡିକୁ ସଂରକ୍ଷିତ ରଖିବାର ଆବଶ୍ୟକତା ନାହିଁ।

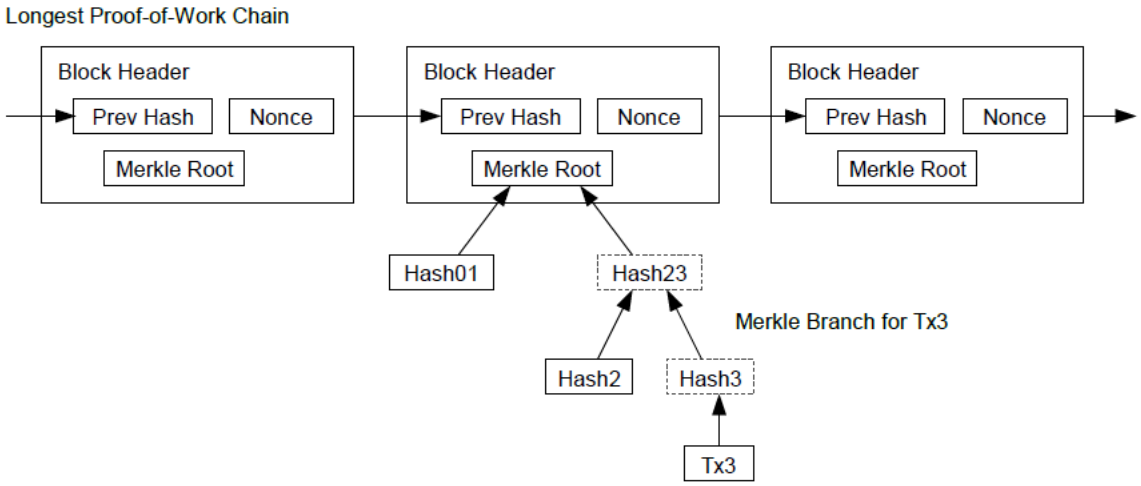
.

ବିନା ଟ୍ରାଞ୍ଜାକ୍ସନରେ ଏକ ବ୍ଲକ୍ ହେଡର୍ ପ୍ରାୟ ୮୦ ବାଇଟ୍ର ହେବ। ଯଦି ମନେ କରନ୍ତୁ ବ୍ଲକ୍ଗୁଡିକୁ ପ୍ରତି ୧୦ ମିନିଟ୍ରେ ଥରେ ସୃଷ୍ଟି ହୁଏ, ତେବେ ପ୍ରତିବର୍ଷ ୮୦ ବାଇଟ୍ *୬*୨୪*୩୬୫= ୪.୨ଏମ୍ବି ହେବ। ୨୦୦୮ ସୁଦ୍ଧା କମ୍ପ୍ୟୁଟର ସିଷ୍ଟମଗୁଡିକୁ ମୁଳତଃ ୨ଜିବି ରାମ୍ ସହ ବିକ୍ରି କରାଯାଉଥିଲା ଏବଂ ମୁରେସ୍ ଲ' ବାର୍ଷିକ ୧.୨ ଜିବିର ବିଦ୍ୟମାନ ଅଭିବୃଦ୍ଧିର ଆକଳନ କରିଛି, ଯଦ ବ୍ଲକ୍ ହେଡର୍ଗୁଡିକୁ ମେମୋରୀ ମଧ୍ୟରେ ରଖାଯାଏ ତେବେ ଷ୍ଟୋରେଜ ଏକ ସମସ୍ୟା ହେବ ନାହିଁ।

୮. ସିମ୍ପ୍ଲିଫାଏଡ ପେମେଂଟ୍ ଭେରିଫିକେଶନ

ଏକ ଫୁଲ୍ ନେଟ୍ୱର୍କ ନୋଡ୍‌କୁ ପରିଚାଳନା ନକରି ପେମେଂଟ୍‌ଗୁଡିକୁ ଯାଂଚ କରିବା ସମ୍ଭବ ଅଟେ। ଜଣେ ଉପଭୋକ୍ତାଙ୍କୁ କେବଳ ଦୀର୍ଘତମ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ଚେନ୍‌ର ବ୍ଲକ୍ ହେଡରଗୁଡିକର ଏକ କପି ରଖିବାର ଆବଶ୍ୟକତା ରହିଛି, ଯାହାକୁ ସେ ତାଙ୍କ ନିକଟରେ ଦୀର୍ଘତମ ଶୃଙ୍ଖଳା ରହିଛି ବୋଲି ସୁନିଶ୍ଚିତ ନହେବା ପର୍ଯ୍ୟନ୍ତ ନେଟ୍ୱର୍କ ନୋଡ୍‌ଗୁଡିକର ସନ୍ଧାନ କରି ପାଇପାରିବେ ଏବଂ ଟାଇମ୍‌ଷ୍ଟାମ୍ପ କରାଯାଇଥିବା ବ୍ଲକ ସହ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଲିଙ୍କ୍ କରି ମର୍‌କେଲ ବ୍ରାଂଚ୍ ହାସଲ କରିପାରିବ। ସେ ନିଜେ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଯାଂଚ କରିପାରିବେ ନାହିଁ କିନ୍ତୁ ଏହାକୁ ଶୃଙ୍ଖଳାର ଏକ ସ୍ଥାନରେ ସଂଯୋଗ କରି ସେ ଦେଖିପାରିବେ ଯେ, ଏକ ନେଟ୍ୱର୍କ ନୋଡ୍ ଏହାକୁ ଗ୍ରହଣ କରିଛି ଏବଂ ଏହାପରେ ଯୋଗ କରାଯାଇଥିବା ବ୍ଲକ୍‌ଗୁଡିକ ନେଟ୍ୱର୍କ ଏହାକୁ ଗ୍ରହଣ କରିଛି ବୋଲି ଅଧିକ ସୁନିଶ୍ଚିତ କରିଥାନ୍ତି।

.



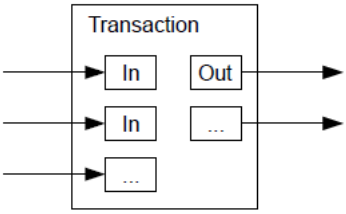
ଏହିପରି ଭାବେ ସଚ୍ଚୋଟ ନୋଡ୍‌ଗୁଡିକ ନେଟ୍ୱର୍କକୁ ନିୟନ୍ତ୍ରଣ କରିବା ପର୍ଯ୍ୟନ୍ତ ଯାଂଚ ନିର୍ଭରଯୋଗ୍ୟ ହୋଇଥାଏ, କିନ୍ତୁ ଯଦି ନେଟ୍ୱର୍କ ଜଣେ ଆକ୍ରମଣକାରୀଙ୍କ ଦ୍ୱାରା ଅଧିକ ଶକ୍ତି ହାସଲ କରେ ତେବେ ଏହା ଅଧିକ ଅସୁରକ୍ଷିତ ହୋଇଥାଏ। ନେଟ୍ୱର୍କ ନୋଡ୍‌ଗୁଡିକ ନିଜେ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ଯାଂଚ କରିପାରୁଥିବା ବେଳେ ସରଳ ପଦ୍ଧତିକୁ ଜଣେ ଆକ୍ରମଣକାରୀଙ୍କ ନକଲି ଟ୍ରାଞ୍ଜାକ୍ସନ ବୋକା ବନାଇପାରେ, ଯେପର୍ଯ୍ୟନ୍ତ ଆକ୍ରମଣକାରୀ ନେଟ୍ୱର୍କଠାରୁ ଅଧିକ ଶକ୍ତିଶାଳୀ ହେବା ଜାରି ରଖିପାରିଥାଏ। ଏହା ବିରୋଧରେ ସୁରକ୍ଷା ପ୍ରଦାନ କରିବା ପାଇଁ ଗୋଟିଏ ରଣନୀତି ହେଉଛି ନେଟ୍ୱର୍କ ନୋଡ୍‌ଗୁଡିକଠାରୁ ଆଲର୍ଟ ଗ୍ରହଣ କରିବା, ଯେତେବେଳେ ସେମାନେ ଏକ ଅବୈଧ ବ୍ଲକ୍ ଚିହ୍ନଟ କରିଥାନ୍ତି, ଉପଭୋକ୍ତାଙ୍କ ସଫ୍ଟୱେୟାରକୁ ଫୁଲ୍ ବ୍ଲକ୍‌କୁ ଡାଉନ୍‌ଲୋଡ କରିବା ପାଇଁ କହିଥାଏ ଏବଂ ଅସାମଞ୍ଜସ୍ୟକୁ ନିଶ୍ଚିତ କରିବା ପାଇଁ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ଆଲର୍ଟ କରିଥାଏ। ବାରମ୍ବାର ପେମେଂଟ୍ ଗ୍ରହଣ କରୁଥିବା ବ୍ୟବସାୟଗୁଡିକ ଅଧିକ ସ୍ୱାଧୀନ ସୁରକ୍ଷା ଓ ଶୀଘ୍ର ଯାଂଚ ପାଇଁ ଏବେ ବି ନିଜସ୍ୱ ନୋଡ୍ ପରିଚାଳନା କରିବାକୁ ଚାହିଁବେ।

.

୯. ମୂଲ୍ୟକୁ ମିଶ୍ରଣ ଓ ବିଭାଜନ କରିବା

ଯଦିଓ କଏନଗୁଡିକୁ ବ୍ୟକ୍ତିଗତ ଭାବେ ପରିଚାଳନା କରିବା ସମ୍ଭବ ଅଟେ, କିନ୍ତୁ ଏକ ଟ୍ରାନ୍ସଫର୍‌ରେ ପ୍ରତ୍ୟେକ କଏନ ପାଇଁ ପୃଥକ ଟ୍ରାଞ୍ଜାକ୍ସନ କରିବା କଷ୍ଟଦାୟକ ହେବ। ମୂଲ୍ୟକୁ ବିଭାଜନ ଓ ମିଶ୍ରଣ କରିବା ପାଇଁ ଅନୁମତି ଦେବା ଲାଗି ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକରେ ଏକାଧିକ ଇନ୍‌ପୁଟ ଓ ଆଉଟପୁଟ୍ ରହିଥାଏ। ସାଧାରଣତଃ ସେଥିରେ ପୂର୍ବର ଏକ ବଡ ଟ୍ରାଞ୍ଜାକ୍ସନରୁ ସିଙ୍ଗଲ୍ ଆଉଟପୁଟ୍ ରହିବ କିମ୍ବା ଛୋଟ ଛୋଟ

ପରିମାଣ ମିଶି ଏକାଧିକ ଇନ୍‌ପୁଟ୍ ରହିବ ଏବଂ ସର୍ବାଧିକ ୨ଟି ଆଉଟପୁଟ ରହିବ: ଗୋଟିଏ ପେମେଂଟ୍ ପାଇଁ ଏବଂ ଯଦି ଗୋଟିଏ ସେଣ୍ଡରଙ୍କ ନିକଟକୁ ଫେରିଯାଏ ତେବେ ଅନ୍ୟଟି ପରିବର୍ତନ ପାଇଁ ଫେରିଥାଏ।.

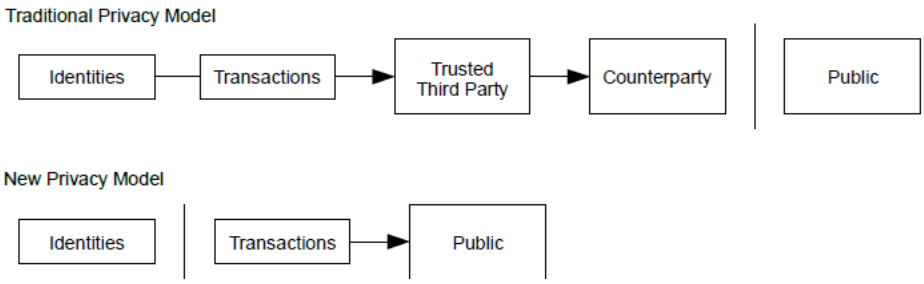


ଏହା ଧ୍ୟାନ ଦେବା ଉଚିତ ଯେ, ଫ୍ୟାନ୍-ଆଉଟ୍, (ଯେଉଁଠାରେ ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନ ଭିନ୍ନ ଭିନ୍ନ ଟ୍ରାଞ୍ଜାକ୍ସନ ଉପରେ ନିର୍ଭର କରିଥାଏ ଏବଂ ଉକ୍ତ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ଆହୁରି ଅନେକ ଟ୍ରାଞ୍ଜାକ୍ସନ ଉପରେ ନିର୍ଭର କରିଥାନ୍ତି) ଏଠାରେ ଏକ ସମସ୍ୟା ନୁହେଁ। ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନର ହିଷ୍ଟ୍ରିର ଏକ ସମ୍ପୂର୍ଣ୍ଣ ଷ୍ଟାଣ୍ଡାଲୋନ କପି ବାହାର କରିବାର କୌଣସି ଆବଶ୍ୟକତା ନାହିଁ।

.

୧୦. ପ୍ରାଇଭେସି (ଗୋପନୀୟତା)

ପାରମ୍ପରିକ ବ୍ୟାଙ୍କିଙ୍ଗ୍ ମଡେଲ ସମ୍ପୃକ୍ତ ଥିବା ପକ୍ଷ ଓ ବିଶ୍ୱସନୀୟ ୩ୟ ପକ୍ଷଗୁଡିକ ପାଇଁ ସୂଚନାର ପହଂଚକୁ ସୀମିତ କରି ଗୋପନୀୟତାର ଏକ ସ୍ତର ହାସଲ କରିଥାଏ। ସମସ୍ତ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ସାର୍ବଜନିନ ଭାବେ ଘୋଷଣା କରିବାର ଆବଶ୍ୟକତା, ଏହି ପଦ୍ଧତିକୁ ବାଦ୍ ଦେଇଥାଏ, କିନ୍ତୁ ଗୋପନୀୟତାକୁ ସାର୍ବଜନିନ ଚାବିଗୁଡିକୁ ଅଜ୍ଞାତ କରି ଅନ୍ୟ ଏକ ସ୍ଥାନରେ ସୂଚନାର ପ୍ରବାହକୁ ଭାଙ୍ଗି ବଜାୟ ରଖାଯାଇପାରିବ। ସାଧାରଣ ଲୋକେ ଦେଖିପାରିବେ ଯେ, ଜଣେ ବ୍ୟକ୍ତି ଅନ୍ୟ ଜଣଙ୍କୁ ଏକ ଅର୍ଥ ରାଶି ପଠାଉଛନ୍ତି କିନ୍ତୁ ସୂଚନା, ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଅନ୍ୟ କୌଣସି ବ୍ୟକ୍ତିଙ୍କ ସହ ସଂଯୋଗ ନକରି। ଏହା ଷ୍ଟକ୍ ଏକ୍ସଚେଞ୍ଜଗୁଡିକ ଦ୍ୱାରା ଜାରି କରାଯାଉଥିବା ସୂଚନାର ସ୍ତର ସହ ସମାନ ଅଟେ, ଯେଉଁଠାରେ ପକ୍ଷଗୁଡିକ କିଏ ସେସମ୍ପର୍କରେ ସୂଚନା ନଦେଇ ବ୍ୟକ୍ତିଗତ କାରବାରଗୁଡିକର ସମୟ ଓ ଆକାର 'ଟେପ୍'କୁ ସାର୍ବଜନିନ କରାଯାଇଥାଏ।



ଏକ ଅତିରିକ୍ତ ଫାୟାରୱାଲ ଭାବେ, ଜଣେ ସାଧାରଣ ମାଲିକଙ୍କ ସହ ସଂଯୋଜିତ ହେବାରୁ ସେଗୁଡିକୁ ସୁରକ୍ଷିତ ରଖିବା ପାଇଁ ଏକ ନୂଆ ଚାବି ପେୟାରକୁ ପ୍ରତ୍ୟେକ ଟ୍ରାଞ୍ଜାକ୍ସନ ପାଇଁ ବ୍ୟବହାର କରାଯିବା ଉଚିତ। ମଲ୍ଟି-ଇନ୍‌ପୁଟ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ସହ କିଛି ସଂଯୋଗକୁ ଏବେ ବି ଏଡା ଯାଇପାରୁନାହିଁ। ବିପଦ ହେଉଛି ଯଦି ଏକ ଚାବିର ମାଲିକଙ୍କ ତଥ୍ୟ ପ୍ରକାଶିତ ହୁଏ ତେବେ ସଂଯୋଗ ଉକ୍ତ ମାଲିକଙ୍କର ଅନ୍ୟ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକୁ ମଧ୍ୟ ପ୍ରକାଶ କରିପାରେ।

୧୧. କାଲ୍‌କୁଲେଶନ୍ସ ବା ଗଣନା

ଏଠାରେ ଆମେ ଜଣେ ଆକ୍ରମଣକାରୀ ବାସ୍ତବ ଶୃଙ୍ଖଳା ତୁଳନାରେ ଦ୍ରୁତ ଥିବା ଏକ ବିକଳ୍ପ ଶୃଙ୍ଖଳା ସୃଷ୍ଟି କରିବା ପାଇଁ ଚେଷ୍ଟା କରୁଥିବା ପରିଦୃଶ୍ୟକୁ ବିଚାରକୁ ନେବା। ଯଦି ବି ସେ ଏଥିରେ ସଫଳ ହୁଅନ୍ତି, ତେବେ ଏହା ସିଷ୍ଟମକୁ ସ୍ୱେଚ୍ଛାକୃତ ପରିବର୍ତନଗୁଡିକ ପାଇଁ ବାଧ୍ୟ କରିନଥାଏ, ଯେପରିକି ଛୋଟିଆ ଫାଙ୍କରୁ ମୂଲ୍ୟ ସୃଷ୍ଟି କରିବା କିମ୍ବା ଅର୍ଥ ହଡପ କରିବା ଭଳି ଯାହା କେବେ ବି ଆକ୍ରମଣକାରୀଙ୍କ ନଥିଲା। ନୋଡ୍‌ଗୁଡିକ ପେମେଂଟ୍ ଭାବେ ଅବୈଧ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଗ୍ରହଣ କରିବେ ନାହିଁ ଏବଂ ବାସ୍ତବ ନୋଡ୍ ଅବୈଧ ଟ୍ରାଞ୍ଜାକ୍ସନ ଥିବା ଏକ ବ୍ଲକ୍‌କୁ କେବେ ବି ଗ୍ରହଣ କରିବେ ନାହିଁ। ଜଣେ ଆକ୍ରମଣକାରୀ କେବଳ ନିକଟରେ ଖର୍ଚ୍ଚ କରିଥିବା ଅର୍ଥକୁ ଫେରି ପାଇବା ପାଇଁ ନିଜର କୌଣସି ଏକ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ପରିବର୍ତନ କରିବା ପାଇଁ ଚେଷ୍ଟା କରିପାରିବେ।

ବାସ୍ତବ ଶୃଙ୍ଖଳା ଓ ଜଣେ ଆକ୍ରମଣକାରୀ ଶୃଙ୍ଖଳା ମଧ୍ୟରେ ପ୍ରତିଦ୍ୱନ୍ଦିତାକୁ ଏକ ବାଇନୋମିଆଲ ର‍ୟାଣ୍ଡମ ୱାକ୍ ଭାବେ ବ୍ୟାଖ୍ୟା କରାଯାଇପାରିବ। ସଫଳତା ଘଟଣା ହେଉଛି ବାସ୍ତବ ଶୃଙ୍ଖଳା ଗୋଟିଏ ବ୍ଲକ୍ ଦ୍ୱାରା ବୃଦ୍ଧି ହେବ, ଯାହା ଏହାର ନେତୃତ୍ୱକୁ ୧+ରେ ଆଗକୁ ବଢାଇଥାଏ ଏବଂ ବିଫଳ ପ୍ରୟାସ ହେଉଛି ଆକ୍ରମଣକାରୀଙ୍କ ଶୃଙ୍ଖଳା ଗୋଟିଏ ବ୍ଲକ୍ ଦ୍ୱାରା ବୃଦ୍ଧି ପାଇଥାଏ ଯାହା ଅନ୍ତରକୁ -୧କୁ ହ୍ରାସ କରିଥାଏ।

ଜଣେ ଆକ୍ରମଣକାରୀ ଏକ ନିର୍ଦ୍ଦିଷ୍ଟ ହ୍ରାସରୁ ପୁଣିଥରେ ଉପରକୁ ଉଠିବାର ସମ୍ଭାବନା ଜଣେ ଗ୍ୟାମ୍ବଲରଙ୍କ ରୁଇନ୍ ସମସ୍ୟା ସହ ସମାନ ଅଟେ। ମନେକର ଅସୀମିତ କ୍ରେଡିଟ୍ ସହ ଜଣେ ଗ୍ୟାମ୍ବଲର୍ ନିଅଂଟରୁ ଆରମ୍ଭ କରନ୍ତି ଏବଂ ବ୍ରେକ୍‌ଇଭେନରେ ପହଂଚିବା ପାଇଁ ଅସୀମ ସଂଖ୍ୟକ ଟ୍ରାଏଲ ପରୀକ୍ଷଣ କରିଥାନ୍ତି। ସେ ବ୍ରେକ୍‌ଇଭେନରେ ପହଂଚିଛନ୍ତି କିମ୍ବା ଜଣେ ଆକ୍ରମଣକାରୀ ସଚ୍ଚୋଟ ଶୃଙ୍ଖଳା ସହ ସମାନ ହୋଇପାରିଛି ତାହାର ସମ୍ଭାବ୍ୟତାକୁ ଆମେ ନିମ୍ନ ଅନୁଯାୟୀ ଗଣନା କରିପାରିବା।

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

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

ଆମର ଆକଳନ ଅନୁଯାୟୀ ଯଦି $p > q$ ହୁଏ, ତେବେ ସମ୍ଭାବନା ବହୁମାତ୍ରାରେ ହ୍ରାସ ପାଇଥାଏ କାରଣ ଆକ୍ରମଣକାରୀଙ୍କ ପାଖରେ ଥିବା ବ୍ଲକ୍‌ର ସଂଖ୍ୟା ବଢାଇବାକୁ ପଡିଥାଏ। ତାଙ୍କ ବିରୋଧରେ ପ୍ରତିବନ୍ଧକ ସତ୍ତ୍ୱେ ଯଦି ସେ ଶୀଘ୍ର ଏକ ଉନ୍ନତ ପଦକ୍ଷେପ ଗ୍ରହଣ ନକରନ୍ତି ତେବେ ତାଙ୍କର ସୁଯୋଗ ଧୀରେ ଧୀରେ ହ୍ରାସ ପାଇଥାଏ କାରଣ ସେ ଆହୁରି ପଛରେ ପଡିଯାଆନ୍ତି।

ବର୍ତମାନ ଆମେ ବିଚାର କରିବା, ସେଣ୍ଡର ବା ପ୍ରେରକ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ପରିବର୍ତନ କରିପାରିବେ ନାହିଁ ବୋଲି ଭଲ ଭାବରେ ନିଶ୍ଚିତ ହେବା ପୂର୍ବରୁ ଏକ ନୂଆ ଟ୍ରାଞ୍ଜାକ୍ସନର ପ୍ରାପ୍ତକର୍ତା (ରେସିପିଏଂଟ)ଙ୍କୁ କେତେ ସମୟ ଅପେକ୍ଷା କରିବାର ଆବଶ୍ୟକତା ରହିଛି। ଆମେ ଅନୁମାନ କରୁଛୁ, ସେଣ୍ଡର ବା ପ୍ରେରକ ହେଉଛନ୍ତି ଜଣେ ଆକ୍ରମଣକାରୀ ଯିଏ ପ୍ରାପ୍ତକର୍ତାଙ୍କୁ ବିଶ୍ୱାସ ଦେବାକୁ ଚାହୁଁଛନ୍ତି ଯେ, ସେ କିଛି ସମୟ ପୂର୍ବରୁ ଟଙ୍କା ପଇଠ କରିଛନ୍ତି, ଏହାପରେ କିଛି ସମୟ ଅତିବାହିତ ହେବା ପରେ ଏହାକୁ ଫେରସ୍ତ କରିବା ପାଇଁ ସେ ସ୍ୱିଚ୍ କରିଥାନ୍ତି। ଯେତେବେଳେ ଏହା ଘଟିବ, ରିସିଭର୍ ବା ଟଙ୍କା ପ୍ରାପ୍ତକର୍ତାଙ୍କୁ ସତର୍କ କରାଯିବ କିନ୍ତୁ ପ୍ରେରକ ଆଶା କରନ୍ତି ଏହା ଖୁବ୍ ବିଳମ୍ବ ହୋଇଯିବ।

ପ୍ରାପ୍ତକର୍ତା ବା ରିସିଭର ଏକ ନୂଆ କୀ ପେୟାର ସୃଷ୍ଟି କରନ୍ତି ଏବଂ ସାଇନିଙ୍ଗ କରିବା ପୂର୍ବରୁ ସାର୍ବଜନିନ କୀ'କୁ ପ୍ରେରକଙ୍କୁ ପ୍ରଦାନ କରନ୍ତି। ଏହା ପ୍ରେରକଙ୍କୁ ଯେପର୍ଯ୍ୟନ୍ତ ସେ ଆଗକୁ ଯିବା ପାଇଁ ଭାଗ୍ୟଶାଳୀ ନହୋଇଥାନ୍ତି, କ୍ରମାଗତ ଭାବେ କାର୍ଯ୍ୟ କରି ସମୟ ପୂର୍ବରୁ ବ୍ଲକ୍‌ଗୁଡିକର ଏକ ଶୃଙ୍ଖଳା ପ୍ରସ୍ତୁତ କରିବା ଓ ପରେ ଉକ୍ତ ମୁହୂର୍ତରେ ଟ୍ରାଞ୍ଜାକ୍ସନକୁ କାର୍ଯ୍ୟକାରୀ କରିବାରୁ ରୋକିଥାଏ। ଥରେ ଟ୍ରାଞ୍ଜାକ୍ସନ ପଠାଯିବା ପରେ, ଅସାଧୁ ପ୍ରେରକ ଏକ ପାରାଲାଲ ଚେନ୍‌ରେ ଗୁପ୍ତ ଭାବେ କାର୍ଯ୍ୟ କରିବା ଆରମ୍ଭ କରନ୍ତି ଯେଉଁଥିରେ ତାଙ୍କ ଟ୍ରାଞ୍ଜାକ୍ସନର ଏକ ବିକଳ୍ପ ଭର୍ସନ ଉପଲବ୍ଧ ଥାଏ।

ପ୍ରାପ୍ତକର୍ତା ଟ୍ରାଞ୍ଜାକ୍ସନକୁ ଏକ ବ୍ଲକ୍‌ରେ ଯୋଗ କରାଯିବା ପର୍ଯ୍ୟନ୍ତ ଅପେକ୍ଷା କରନ୍ତି ଏବଂ ଜେଡ୍ ବ୍ଲକ୍‌ଗୁଡିକୁ ଏହାପରେ ଲିଙ୍କ୍ କରାଯାଇଥାଏ। ସେ ଆକ୍ରମଣକାରୀ କେତେ ପରିମାଣର ଅର୍ଥ ହାସଲ କରଛନ୍ତି ତାହା ଜାଣିନାହାନ୍ତି, କିନ୍ତୁ ଅନୁମାନ କରନ୍ତି ଯେ, ସଚ୍ଚୋଟ ବ୍ଲକ୍‌ଗୁଡିକ, ପ୍ରତି ବ୍ଲକ୍ ପିଛା ହାରାହାରି ଆକାଂକ୍ଷିତ ସମୟ ନେଇଛନ୍ତି, ଆକ୍ରମଣକାରୀଙ୍କ ସମ୍ଭାବ୍ୟ ପ୍ରଗତି ଆକାଂକ୍ଷିତ ମୂଲ୍ୟ ସହ ଏକ ପଏସନ ଡିଷ୍ଟ୍ରିବ୍ୟୁଶନ ହେବ

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

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
```

୧୨. କନ୍‌କ୍ଲୁଜନ (ସିଦ୍ଧାନ୍ତ)

ଆମେ ବିଶ୍ୱାସ ଉପରେ ନିର୍ଭର ନକରି ଇଲେକ୍‌ଟ୍ରୋନିକ ଟ୍ରାଞ୍ଜାକ୍ସନଗୁଡିକ ପାଇଁ ଏକ ସିଷ୍ଟମର ପ୍ରସ୍ତାବ ଦେଇଛୁ। ଆମେ ଡିଜିଟାଲ ସିଗ୍ନେଚରଗୁଡିକରୁ ନିର୍ମିତ କଏନଗୁଡିକର ଏକ ସାଧାରଣ ଫ୍ରେମ୍‌ୱର୍କ ସହ ଆରମ୍ଭ କରିଛୁ, ଯାହା ମାଲିକାନାର ମଜବୁତ ନିୟନ୍ତ୍ରଣ ପ୍ରଦାନ କରୁଛି କିନ୍ତୁ ଏହା ଦୁଇ ଗୁଣ ବ୍ୟୟକୁ ରୋକିବା ପାଇଁ ଏକ ଉପାୟ ବିନା ଅସମ୍ପୂର୍ଣ୍ଣ ଅଟେ। ଏହାର ସମାଧାନ ପାଇଁ ଆମେ ଟ୍ରାଞ୍ଜାକ୍ସନର ଏକ ସାର୍ବଜନିନ ହିଷ୍ଟ୍ରି ରେକର୍ଡ କରିବା ଲାଗି ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କର ବ୍ୟବହାର କରି ଏକ ପିୟର-ଟୁ-ପିୟର ନେଟ୍‌ୱର୍କର ପ୍ରସ୍ତାବ ଦେଇଛୁ ଯାହା ଜଣେ ଆକ୍ରମଣକାରୀଙ୍କ ପାଇଁ ପରିବର୍ତନ ହେବା ଲାଗି ଶୀଘ୍ର ଗଣନାତ୍ମକ ଭାବେ ଅସାଧ୍ୟ ହୋଇଥାଏ, ଯଦି ସଚ୍ଚୋଟ ନୋଡ୍‌ଗୁଡିକ ଅଧିକାଂଶ ସିପିୟୁ ପାୱାରକୁ ନିୟନ୍ତ୍ରଣ କରନ୍ତି। ଏହି ନେଟ୍‌ୱର୍କ ନିଜ ଅଣସଂଗଠିତ ସରଳତା ମଧ୍ୟରେ ମଜବୁତ ଅଟେ। ନୋଡ୍‌ଗୁଡିକ ସାମାନ୍ୟ ସମନ୍ୱୟ ସହିତ ଏକାସଙ୍ଗେ କାର୍ଯ୍ୟ କରିଥାନ୍ତି। ସେମାନଙ୍କୁ ଚିହ୍ନଟ କରିବାର ଆବଶ୍ୟକତା ନହିଁ, ଯେହେତୁ ବାର୍ତାଗୁଡିକ କୌଣସି ନିର୍ଦ୍ଦିଷ୍ଟ ସ୍ଥାନକୁ ଯାଇନଥାଏ ଏବଂ ସେଗୁଡିକୁ କେବଳ ଏକ ସର୍ବୋତମ ପ୍ରୟାସ ଆଧାରରେ ବିତରଣ କରାଯିବାର ଆବଶ୍ୟକତା ରହିଛି। ନୋଡ୍‌ଗୁଡିକ ନିଜ ଇଚ୍ଛାରେ ନେଟ୍‌ୱର୍କକୁ ଛାଡିପାରିବେ ଓ ପୁନଃ ଯୋଗ ଦେଇପାରିବ, ସେମାନେ ଯିବା ପରେ କ'ଣ ଘଟିଥିଲା ତାହାର ପ୍ରମାଣ ଭାବେ ପ୍ରୁଫ୍-ଅଫ୍-ୱର୍କ ଶୃଙ୍ଖଳାକୁ ଗ୍ରହଣ କରିପାରନ୍ତି। ସେମାନେ ନିଜର ସିପିୟୁ ପାୱାର ସହ ଭୋଟ ଦେଇପାରିବେ, ସେଗୁଡିକୁ

ବିସ୍ତାର କରିବା ଉପରେ କାର୍ଯ୍ୟ କରି ବୈଧ ବ୍ଲକ୍‌ଗୁଡିକୁ ଗ୍ରହଣ କରିପାରିବେ ଏବଂ ଅବୈଧ ବ୍ଲକ୍‌ଗୁଡିକୁ ସେମାନଙ୍କ ସହ କାର୍ଯ୍ୟ କରିବା ପାଇଁ ମନା କରି ପ୍ରତ୍ୟାଖ୍ୟାନ କରିପାରିବେ। ଯେକୌଣସି ଆବଶ୍ୟକୀୟ ନିୟମ ଓ ପ୍ରୋତ୍ସାହନକୁ ଏହି ସର୍ବସମ୍ମତ ପ୍ରଣାଳୀ ସହିତ

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.