ಬಿಟ್ಕಾಯಿನ್: ಪೀರ್-ಟು-ಪೀರ್ ಎಲೆಕ್ಟ್ರಾನಿಕ್ ಕ್ಯಾಶ್ ಸಿಸ್ಟಂ

ಸತೋಷಿ ನಕಮೊಟೊ
satoshin@gmx.com
www.bitcoin.org

ಸಾರಾಂಶ. ಎಲೆಕ್ಟ್ರಾನಿಕ್ ನಗದಿನ ಶುದ್ಧ ಪೀರ್-ಟು-ಪೀರ್ ಆವೃತ್ತಿಯು ಆನ್ಲೈನ್ ಪಾವತಿಗಳನ್ನು ಹಣಕಾಸು ಸಂಸ್ಥೆಯ ಮೂಲಕ ಹೋಗದೆ ನೇರವಾಗಿ ಒಬ್ಬರಿಂದ ಮತ್ತೊಬ್ಬರಿಗೆ ಕಳುಹಿಸುವುದಕ್ಕೆ ಅವಕಾಶ ನೀಡುತ್ತದೆ. ಡಿಜಿಟಲ್ ಸಹಿಗಳು ಪರಿಹಾರದ ಭಾಗವನ್ನು ನೀಡುತ್ತವೆ, ಆದರೆ ಎರಡು ಬಾರಿ ವ್ಯಯಿಸುವುದನ್ನು ತಪ್ಪಿಸಲು ವಿಶ್ವಾಸಾರ್ಹ ಮೂರನೇ ಪಕ್ಷದ ಅಗತ್ಯವಿದೆ. ನಾವು ಪೀರ್-ಟು-ಪೀರ್ ಜಾಲವನ್ನು ಬಳಸಿ ಎರಡು ನಾರಿ ವ್ಯಯಿಸುವ ಸಮಸ್ಯೆಗೆ ಪರಿಹಾರವನ್ನು ಪ್ರಸ್ತಾವಿಸಿದ್ದೇವೆ. ನೆಟ್ವರ್ಕ್ ಟೈಂಸ್ಟಾಂಪ್ಸ್ ವಹಿವಾಟುಗಳನ್ನು ಅವುಗಳನ್ನು ಹ್ಯಾಶ್-ಆಧರಿತ ಪ್ರೂಫ್-ಆಫ್- ವರ್ಕ್ ನೆಟ್ವರ್ಕ್ ಗೆ ಹ್ಯಾಶಿಂಗ್ ಮಾಡುವ ಮೂಲಕ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಪುನಃ ಮಾಡದೆ ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲದ ದಾಖಲೆ ರೂಪಿಸುತ್ತದೆ. ಈ ಉದ್ದದ ಸರಪಳಿಯು ಘಟನೆಗಳ ಸರಣಿಯ ಪುರಾವೆಯಾಗಿ ಮಾತ್ರ ಉಳಿಯುವುದಲ್ಲದೆ ಸಿಪಿಯು ಶಕ್ತಿಯ ಅತ್ಯಂತ ದೊಡ್ಡ ರಾಶಿಯ ಮೂಲಕ ಬರುತ್ತದೆ. ಬಹಳಷ್ಟು ಸಿಪಿಯು ಶಕ್ತಿಯನ್ನು ನೋಡ್ ಗಳಿಂದ ನಿಯಂತ್ರಿಸುವವರೆಗೆ ಅದು ನೆಟ್ವರ್ಕ್ ಗೆ ದಾಳಿ ಮಾಡಲು ಸಹಕರಿಸುವುದಿಲ್ಲ, ಅವು ಅತ್ಯಂತ ಉದ್ದದ ಸರಪಳಿ ಸೃಷ್ಟಿಸುತ್ತವೆ ಮತ್ತು ದಾಳಿಕೋರರನ್ನು ಮೀರಿಸುತ್ತವೆ. ನೆಟ್ವರ್ಕ್ ಗೆ ಸ್ವತಃ ಕನಿಷ್ಠಾತ್ಮಕ ರಚನೆ ಅಗತ್ಯವಾಗುತ್ತದೆ. ಅತ್ಯುತ್ತಮ ಪ್ರಯತ್ನ ಆಧರಿತವಾಗಿ ಪ್ರಸಾರವಾಗುವ ಸಂದೇಶಗಳು ಮತ್ತು ನೋಡ್ ಗಳು ನೆಟ್ವರ್ಕ್ ಅನ್ನು ಅದಕ್ಕೆ ಇಷ್ಟಬಂದಂತೆ ತ್ಯಜಿಸಬಹುದು ಮತ್ತು ಮರು ಸೇರಿಕೊಳ್ಳುವ ಮೂಲಕ ಅತ್ಯಂತ ಉದ್ದದ ಪ್ರೂಫ್-ಆಫ್ ವರ್ಕ್ ಸರಪಳಿಯನ್ನು ಅದನ್ನು ಅವರು ಹೊರಗಿದ್ದಾಗ ಏನಾಗಿದೆ ಎನ್ನುವುದಕ್ಕೆ ಸಾಕ್ಷಿಯಾಗಿ ಅನುಮೋದಿಸಬಹುದು.

1. ಪ್ರಸ್ತಾವನೆ

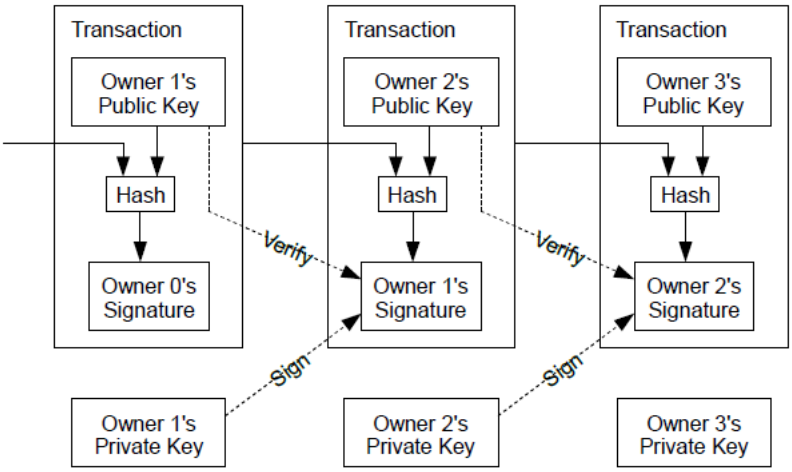
ಅಂತರ್ಜಾಲದಲ್ಲಿ ವಾಣಿಜ್ಯ ಎನ್ನುವುದು ವಿಶೇಷವಾಗಿ ಎಲೆಕ್ಟ್ರಾನಿಕ್ ಪಾವತಿಗಳ ಸಂಸ್ಕರಣೆಗೆ ವಿಶ್ವಾಸಾರ್ಹ ಮೂರನೇ ಪಕ್ಷಗಳು ಹಣಕಾಸು ಸಂಸ್ಥೆಗಳು ವಿಶೇಷವಾಗಿ ಸೇವೆ ಒದಗಿಸುವುದರಲ್ಲಿ ನಂಬಿಕೆ ಇರಿಸಿವೆ. ಈ ವ್ಯವಸ್ಥೆಯು ಬಹಳಷ್ಟು ವಹಿವಾಟುಗಳಿಗೆ ಚೆನ್ನಾಗಿ ಕೆಲಸ ಮಾಡುತ್ತಿದ್ದರೂ ಅದು ವಿಶ್ವಾಸ ಆಧರಿತ ಮಾದರಿಯಾಗಿ ಆಂತರಿಕ ದೌರ್ಬಲ್ಯದಿಂದ ಇನ್ನೂ ನರಳುತ್ತಿದೆ. ಸಂಪೂರ್ಣ ಹಿಂದಿರುಗಿಸಲಾದ ವಹಿವಾಟುಗಳು ನಿಜಕ್ಕೂ ಸಾಧ್ಯವಿಲ್ಲ, ಹಣಕಾಸು ಸಂಸ್ಥೆಗಳು ವಿವಾದಗಳಿಗೆ ಮಧ್ಯಪ್ರವೇಶವನ್ನು ತಪ್ಪಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ಮಧ್ಯಸ್ಥಿಕೆಯ ವೆಚ್ಚವು ವಹಿವಾಟಿನ ವೆಚ್ಚಗಳನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ, ಕನಿಷ್ಠ ಪ್ರಾಯೋಗಿಕ ವಹಿವಾಟಿನ ಗಾತ್ರ ಮತ್ತು ಸಣ್ಣ ಅನೌಪಚಾರಿಕ ವಹಿವಾಟುಗಳ ಸಾಧ್ಯತೆಯನ್ನು ನಿವಾರಿಸುತ್ತದೆ, ಮತ್ತು ಹಿಂದಿರುಗಿಸಲಾಗದ ಸೇವೆಗಳಿಗೆ ಹಿಂದಿರುಗಿಸಲಾಗದ ಪಾವತಿಗಳನ್ನು ಮಾಡಲು ಸಾಮರ್ಥ್ಯದ ನಷ್ಟದ ವಿಸ್ತಾರ ವೆಚ್ಚವಿರುತ್ತದೆ. ಹಿಮ್ಮುಖದ ಸಾಧ್ಯತೆಯಿಂದ ವಿಶ್ವಾಸದ ಅಗತ್ಯ ವಿಸ್ತರಿಸುತ್ತದೆ. ವ್ಯಾಪಾರಿಗಳು ಅವರ ಗ್ರಾಹಕರ ಬಗ್ಗೆ ಜಾಗರೂಕರಾಗಿರಬೇಕು, ಅವರಿಗೆ ಅಗತ್ಯಕ್ಕಿಂತ ಹೆಚ್ಚಿನ ಮಾಹಿತಿಗಾಗಿ ತೊಂದರೆ ಕೊಡಬಾರದು. ಸ್ವಲ್ಪ ಪ್ರಮಾಣದ

ವಂಚನೆಯನ್ನು ತಪ್ಪಿಸಲಾಗದು ಎಂದು ಒಪ್ಪಿಕೊಳ್ಳಲಾಗುತ್ತದೆ. ಈ ವೆಚ್ಚಗಳು ಮತ್ತು ಪಾವತಿಯ ಅನಿಶ್ಚಿತತೆಗಳನ್ನು ಭೌತಿಕ ಕರೆನ್ಸಿ ಬಳಸಿ ವೈಯಕ್ತಿಕವಾಗಿ ತಪ್ಪಿಸಬಹುದು, ಆದರೆ ಯಾವುದೇ ವ್ಯವಸ್ಥೆಯೂ ವಿಶ್ವಾಸಾರ್ಹ ವ್ಯಕ್ತಿಯಿಲ್ಲದೆ ಸಂವಹನ ಮಾರ್ಗದ ಮೂಲಕ ಪಾವತಿಗಳನ್ನು ಮಾಡುವುದಿಲ್ಲ.

ಇಲ್ಲಿ ಅಗತ್ಯವಿರುವುದು ವಿಶ್ವಾಸಕ್ಕೆ ಕ್ರಿಪ್ಟೊಗ್ರಾಫಿಕ್ ಸಾಕ್ಷ್ಯ ಬದಲಾಗಿ ಎಲೆಕ್ಟ್ರಾನಿಕ್ ಪೇಮೆಂಟ್ ಸಿಸ್ಟಂ ಆಗಿದ್ದು ಅದು ಯಾವುದೇ ಇಬ್ಬರು ಒಪ್ಪಿಗೆ ಇರುವ ವ್ಯಕ್ತಿಗಳ ನಡುವೆ ವಿಶ್ವಾಸಾರ್ಹ ಮೂರನೇ ವ್ಯಕ್ತಿಯ ಅಗತ್ಯವಿಲ್ಲದೆ ಪರಸ್ಪರ ನೇರವಾಗಿ ವಹಿವಾಟು ನಡೆಸುವುದಕ್ಕೆ ಅವಕಾಶ ಕಲ್ಪಿಸುತ್ತದೆ. ಕಂಪ್ಯೂಟೇಷನ್ ಮೂಲಕ ಹಿಂದಿರುಗಿಸಲು ಪ್ರಾಯೋಗಿಕವಲ್ಲದ ವಹಿವಾಟುಗಳು ಮಾರಾಟಗಾರರನ್ನು ವಂಚನೆಯಿಂದ ರಕ್ಷಿಸಬಲ್ಲವು ಮತ್ತು ಖರೀದಿದಾರರನ್ನು ರಕ್ಷಿಸಲು ದಿನನಿತ್ಯದ ಎಸ್ಕ್ರೋ ವ್ಯವಸ್ಥೆಗಳನ್ನು ಸುಲಭವಾಗಿ ಅನುಷ್ಠಾನಗೊಳಿಸಬಹುದು. ಈ ಪ್ರಬಂಧದಲ್ಲಿ ನಾವು ಎರಡು ಬಾರಿ ವ್ಯಯಿಸುವ ಸಮಸ್ಯೆ ಪರಿಹರಿಸಲು ಪೀರ್-ಟು-ಪೀರ್ ವಿತರಿಸಲ್ಪಟ್ಟ ಟೈಂಸ್ಟಾಂಪ್ ಸರ್ವರ್ ಬಳಸಿ ವಹಿವಾಟುಗಳ ಕಾಲಾನುಕ್ರಮದ ಶ್ರೇಣಿಯ ಕಂಪ್ಯೂಟೇಷನಲ್ ಪ್ರೂಫ್ ಸೃಷ್ಟಿಸುವ ಪರಿಹಾರವನ್ನು ಪ್ರಸ್ತಾವಿಸಿದ್ದೇವೆ. ಈ ವ್ಯವಸ್ಥೆಯು ಯಾವುದೇ ಸಹಕರಿಸುವ ದಾಳಿಕೋರರ ನೋಡ್ ಗಳಿಗಿಂತ ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳು ಹೆಚ್ಚು ಸಿಪಿಯು ಶಕ್ತಿಯನ್ನು ನಿಯಂತ್ರಿಸುತ್ತಿರುವಷ್ಟು ಸಮಯವೂ ಸುರಕ್ಷಿತವಾಗಿರುತ್ತದೆ.

2. ವಹಿವಾಟುಗಳು

ನಾವು ಎಲೆಕ್ಟ್ರಾನಿಕ್ ಕಾಯಿನ್ ಅನ್ನು ಡಿಜಿಟಲ್ ಸಿಗ್ನೇಚರ್ ಗಳ ಮೂಲಕ ವ್ಯಾಖ್ಯಾನಿಸುತ್ತೇವೆ. ಪ್ರತಿ ಮಾಲೀಕ ಕಾಯಿನ್ ಅನ್ನು ಹಿಂದಿನ ವಹಿವಾಟಿನ ಹ್ಯಾಶ್ ಅನ್ನು ಡಿಜಿಟಲಿ ಸಹಿ ಮಾಡುವ ಮೂಲಕ ಮತ್ತು ಮುಂದಿನ ಮಾಲೀಕ ಪಬ್ಲಿಕ್ ಕೀ ಮೂಲಕ ವರ್ಗಾಯಿಸುತ್ತಾನೆ ಮತ್ತು ಇವುಗಳನ್ನು ಕಾಯಿನ್ ಅಂತ್ಯದಲ್ಲಿ ಸೇರಿಸುತ್ತಾನೆ. ಹಣ ಪಡೆದವರು ಈ ಸಹಿಗಳನ್ನು ಪರಿಶೀಲಿಸುವ ಮೂಲಕ ಮಾಲೀಕತ್ವದ ಸರಣಿಯನ್ನು ಪರಿಶೀಲಿಸಬಹುದು.



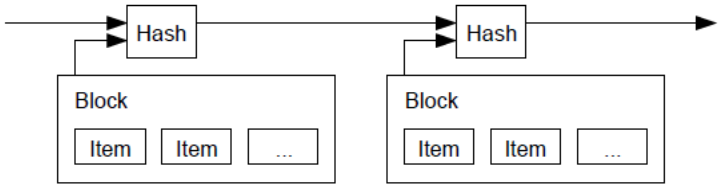
ಈ ಚಲನೆಯಲ್ಲಿನ ಸಮಸ್ಯೆ ಎಂದರೆ ಹಣ ಸ್ವೀಕರಿಸುವವನು ಕಾಯಿನ್ ಅನ್ನು ಎರಡು ಬಾರಿ ವ್ಯಯಿಸಿಲ್ಲ ಎಂದು ದೃಢಪಡಿಸಿಕೊಳ್ಳಲು ಸಾಧ್ಯವಿಲ್ಲ. ಅದಕ್ಕೆ ಸಾಮಾನ್ಯ ಪರಿಹಾರವೆಂದರೆ ವಿಶ್ವಾಸಾರ್ಹ ಕೇಂದ್ರ ಸಂಸ್ಥೆಯೊಂದನ್ನು ಅಥವಾ ಟಂಕಿಸುವ ಸಂಸ್ಥೆಯನ್ನು ತಂದರೆ ಅದು ಪ್ರತಿ ವಹಿವಾಟನ್ನೂ ಎರಡು ಬಾರಿ ವ್ಯಯಿಸಿಲ್ಲ ಎಂದು ಪರೀಕ್ಷಿಸಿಕೊಳ್ಳುತ್ತದೆ. ಪ್ರತಿ ವಹಿವಾಟಿನ ನಂತರ ಕಾಯಿನ್ ಅನ್ನು ಟಂಕಸಾಲೆಗೆ ಹಿಂದಿರುಗಿಸುವ ಮೂಲಕ ಹೊಸ ನಾಣ್ಯ ನೀಡುವಂತೆ ಮಾಡಬೇಕು ಮತ್ತು ಟಂಕಿಸುವ ಸಂಸ್ಥೆ ನೇರವಾಗಿ ನೀಡುವ ನಾಣ್ಯಗಳು ಮಾತ್ರ ಎರಡು ಬಾರಿ ವ್ಯಯಿಸಿಲ್ಲ

ಎನ್ನುವುದಕ್ಕೆ ವಿಶ್ವಾಸಾರ್ಹವಾಗುತ್ತವೆ. ಈ ಪರಿಹಾರದಲ್ಲಿನ ಸಮಸ್ಯೆ ಎಂದರೆ ಪ್ರತಿ ವಹಿವಾಟು ಕೂಡಾ ಅವರಿಂದ ನಡೆಯಬೇಕಾದ್ದರಿಂದ ಬ್ಯಾಂಕಿನಂತೆಯೇ ಇಡೀ ಹಣದ ವ್ಯವಸ್ಥೆಯು ಈ ಟಂಕಿಸುವ ಕಂಪನಿಯನ್ನು ನಡೆಸುವವರ ಮೇಲೆ ಆಧಾರಪಡುತ್ತದೆ.

ನಾವು ಹಣ ಸ್ವೀಕರಿಸುವವರಿಗೆ ಹಿಂದಿನ ಮಾಲೀಕರು ಯಾವುದೇ ಹಿಂದಿನ ವಹಿವಾಟುಗಳನ್ನು ನಡೆಸಿಲ್ಲ ಎಂದು ಅರಿತುಕೊಳ್ಳಲು ಒಂದು ಮಾರ್ಗ ಬೇಕಾಗಿದೆ. ನಮ್ಮ ಉದ್ದೇಶಗಳಿಗೆ ಅತ್ಯಂತ ಪ್ರಾರಂಭಿಕ ವಹಿವಾಟು ಲೆಕ್ಕವಿರುತ್ತದೆ, ಆದ್ದರಿಂದ ನಾವು ನಂತರದ ಪ್ರಯತ್ನಗಳಲ್ಲಿ ಎರಡು ಬಾರಿ ವ್ಯಯಿಸಿದರೆ ಲೆಕ್ಕಿಸಬೇಕಿಲ್ಲ. ವಹಿವಾಟಿನ ಗೈರುಹಾಜರಿಯನ್ನು ದೃಢೀಕರಿಸಿಕೊಳ್ಳಲು ಎಲ್ಲ ವಹಿವಾಟುಗಳ ಕುರಿತೂ ಅರಿವನ್ನು ಹೊಂದುವುದು. ಟಂಕಸಾಲೆ ಆಧರಿತ ಮಾದರಿಯಲ್ಲಿ ಟಂಕಸಾಲೆಗೆ ಎಲ್ಲ ವಹಿವಾಟುಗಳ ಬಗ್ಗೆ ತಿಳಿದಿರುತ್ತದೆ ಮತ್ತು ಯಾವುದು ಮೊದಲು ಬಂದಿದೆ ಎಂದು ನಿರ್ಧರಿಸುತ್ತದೆ. ಇದನ್ನು ವಿಶ್ವಾಸಾರ್ಹ ಪಕ್ಷವಿಲ್ಲದೆ ನೆರವೇರಿಸಲು, ವಹಿವಾಟುಗಳನ್ನು ಸಾರ್ವಜನಿಕವಾಗಿ ಪ್ರಕಟಿಸಬೇಕು [1], ಮತ್ತು ಭಾಗವಹಿಸುವವರು ಅವರು ಸ್ವೀಕರಿಸಿದ ಅನುಕ್ರಮಣಿಕೆಯ ಒಂದು ಇತಿಹಾಸಕ್ಕೆ ಒಪ್ಪುವ ವ್ಯವಸ್ಥೆ ನಮಗೆ ಅಗತ್ಯ. ಹಣ ಪಡೆಯುವವರಿಗೆ ಪ್ರತಿ ವಹಿವಾಟಿ ಸಮಯದಲ್ಲಿ ಸಾಕ್ಷಿ ಬೇಕಾಗುತ್ತದೆ, ಬಹಳಷ್ಟು ನೋಡ್ ಗಳು ಅದನ್ನು ಮೊದಲು ಸ್ವೀಕರಿಸಲಾಗಿದೆ ಎಂದು ಒಪ್ಪಿರಬೇಕು.

3. ಟೈಂಸ್ಟಾಂಪ್ ಸರ್ವರ್

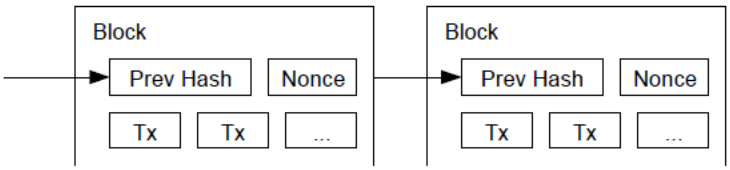
ನಾವು ಪ್ರಸ್ತಾವಿಸುವ ಪರಿಹಾರವು ಟೈಂಸ್ಟಾಂಪ್ ಸರ್ವರ್ ನೊಂದಿಗೆ ಇದೆ. ಟೈಂಸ್ಟಾಂಪ್ ಸರ್ವರ್ ಟೈಂಸ್ಟಾಂಪ್ ಮಾಡಬೇಕಾದ ವಸ್ತುಗಳ ಬ್ಲಾಕ್ ನ ಹ್ಯಾಶ್ ಅನ್ನು ತೆಗೆದುಕೊಂಡು ಕೆಲಸ ಮಾಡುತ್ತದೆ ಮತ್ತು ಹ್ಯಾಶ್ ಅನ್ನು ಎಲ್ಲೆಡೆ ಅಂದರೆ ದಿನಪತ್ರಿಕೆ ಅಥವಾ ಯೂಸ್ ನೆಟ್ ಪೋಸ್ಟ್ [2-5]ಪ್ರಕಟಿಸುತ್ತದೆ. ಟೈಂಸ್ಟಾಂಪ್ ಹ್ಯಾಶ್ ಒಳಗಡೆ ಪ್ರವೇಶಿಸಲು ಆ ಕಾಲದಲ್ಲಿ ಡೇಟಾ ಇತ್ತು ಎಂದು ಸಾಬೀತುಪಡಿಸುತ್ತದೆ. ಪ್ರತಿ ಟೈಂಸ್ಟಾಂಪ್ ತನ್ನ ಹ್ಯಾಶ್ ನಲ್ಲಿನ ಹಿಂದಿನ ಟೈಂಸ್ಟಾಂಪ್ ಒಳಗೊಳ್ಳುತ್ತದೆ, ಇದರಿಂದ ಸರಣಿ ರೂಪುಗೊಳ್ಳುತ್ತದೆ, ಪ್ರತಿ ಹೆಚ್ಚುವರಿ ಟೈಂಸ್ಟಾಂಪ್ ಅದರ ಹಿಂದಿನದನ್ನು ಬಲಪಡಿಸುತ್ತದೆ.



4. ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್

ವಿತರಣೆಯಾದ ಟೈಂಸ್ಟಾಂಪ್ ಸರ್ವರ್ ಅನ್ನು ಪೀರ್-ಟು-ಪೀರ್ ಆಧಾರದಲ್ಲಿ ಅನುಷ್ಠಾನಗೊಳಿಸಲು ನಾವು ಆಡ ಬ್ಯಾಕ್ ನ ಹ್ಯಾಶ್ ಕ್ಯಾಶ್ [6] ರೀತಿಯಂತೆಯೇ ದಿನಪತ್ರಿಕೆ ಅಥವಾ ಯೂಸ್ ನೆಟ್ ಪೋಸ್ಟ್ ಗಳ ಬದಲಿಗೆ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಸಿಸ್ಟಂ ಬಳಸಬೇಕು. ಈ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಹ್ಯಾಶ್ ಮಾಡಿದಾಗ ಮೌಲ್ಯಕ್ಕೆ ಸ್ಕ್ಯಾನಿಂಗ್ ಒಳಗೊಂಡಿರುತ್ತದೆ, ಅಂದರೆ ಎಸ್.ಎಚ್.ಎ-256, ಹ್ಯಾಶ್ ಹಲವಾರು ಶೂನ್ಯ ಬಿಟ್ಸ್ ನೊಂದಿಗೆ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಅಗತ್ಯವಿರುವ ಸರಾಸರಿ ಕೆಲಸವು ಸೊನ್ನೆಯ ಬಿಟ್ಸ್ ಸಂಖ್ಯೆಯಲ್ಲಿ ಘಾತೀಯವಾಗಿರುತ್ತದೆ ಮತ್ತು ಅದನ್ನು ಒಂದು ಹ್ಯಾಶ್ ಅನುಷ್ಠಾನಗೊಳಿಸಿ ಪರಿಶೀಲಿಸಬಹುದು.

ನಮ್ಮ ಟೈಂಸ್ಟಾಂಪ್ ಜಾಲಕ್ಕೆ ನಾವು ಅಗತ್ಯವಿರುವ ಶೂನ್ಯ ಬಿಟ್ಸ್ ಬ್ಲಾಕ್ ನ ಹ್ಯಾಶ್ ನೀಡುವವರೆಗೆ ಬ್ಲಾಕ್ ನಲ್ಲಿ ನಾನ್ಸ್ ಹೆಚ್ಚಿಸುವ ಮೂಲಕ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಅನುಷ್ಠಾನಗೊಳಿಸುತ್ತೇವೆ. ಒಮ್ಮೆ ಸಿಪಿಯು ಪ್ರಯತ್ನವನ್ನು ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಸಂತೃಪ್ತಗೊಳಿಸಲು ವ್ಯಯಿಸಿದ ನಂತರ ಕೆಲಸವನ್ನು ಮತ್ತೆ ಮಾಡದೆ ಬ್ಲಾಕ್ ಅನ್ನು ಬದಲಾಯಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ. ನಂತರ ಬ್ಲಾಕ್ ಗಳನ್ನು ನಂತರ ಸರಪಳಿ ಮಾಡಲಾಗುವ ಮೂಲಕ ಬ್ಲಾಕ್ ಬದಲಾಯಿಸುವ ಕೆಲಸವು ಎಲ್ಲ ಬ್ಲಾಕ್ ಗಳನ್ನೂ ಅದರ ನಂತರ ಪುನಃ ಮಾಡಬೇಕಾಗುತ್ತದೆ.



ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಬಹಳಷ್ಟು ನಿರ್ಧಾರ ಕೈಗೊಳ್ಳುವಲ್ಲಿ ಪ್ರಾತಿನಿಧ್ಯ ನಿರ್ಧರಿಸುವ ಸಮಸ್ಯೆಯನ್ನು ಕೂಡಾ ಪರಿಹರಿಸುತ್ತದೆ. ಬಹಳಷ್ಟು ಮಂದಿ ಒಂದು-ಐಪಿ-ವಿಳಾಸ-ಒಂದು-ಮತ ಆಧರಿಸಿದ್ದರೆ ಅದನ್ನು ಹಲವು ಐಪಿಗಳನ್ನು ನೀಡುವ ಮೂಲಕ ಯಾರೇ ಆದರೂ ಬುಡಮೇಲು ಮಾಡಬಹುದು. ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಮೂಲಭೂತವಾಗಿ ಒಂದು-ಸಿಪಿಯು-ಒಂದು-ಮತವಾಗಿರುತ್ತದೆ. ಬಹಳಷ್ಟು ನಿರ್ಧಾರವನ್ನು ಅತ್ಯಂತ ಉದ್ದದ ಸರಪಳಿಯಿಂದ ಪ್ರತಿನಿಧಿಸಲಾಗುತ್ತದೆ, ಅದು ಅದರಲ್ಲಿ ಹೂಡಿಕೆ ಮಾಡಿದ ಅತ್ಯಂತ ಮಹತ್ತರ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಇರುತ್ತದೆ. ಬಹಳಷ್ಟು ಸಿಪಿಯು ಶಕ್ತಿಯನ್ನು ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳಿಂದ ನಿಯಂತ್ರಿಸಲ್ಪಟ್ಟರೆ ಪ್ರಾಮಾಣಿಕ ಸರಪಳಿಯು ಅತ್ಯಂತ ವೇಗವಾಗಿ ಬೆಳೆಯುತ್ತದೆ ಮತ್ತು ಯಾವುದೇ ಕಂಪ್ಯೂಟಿಂಗ್ ಸರಪಳಿಯನ್ನು ಮೀರುತ್ತದೆ. ಹಿಂದಿನ ಬ್ಲಾಕ್ ಅನ್ನು ಬದಲಾಯಿಸಲು ದಾಳಿಕೋರನು ಬ್ಲಾಕ್ ಮತ್ತು ಮತ್ತು ನಂತರದ ಎಲ್ಲ ಬ್ಲಾಕ್ ಗಳ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಅನ್ನು ಮತ್ತೆ ಮಾಡಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ನಂತರ ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳನ್ನು ಹಿಡಿದುಕೊಳ್ಳಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಮೀರಬೇಕಾಗುತ್ತದೆ. ನಂತರ ನಾವು ನಿಧಾನಗತಿಯ ದಾಳಿಕೋರ ಹಿಡಿದುಕೊಳ್ಳುವ ಸಂಭವನೀಯತೆಯು ನಂತರದ ಬ್ಲಾಕ್ ಗಳನ್ನು ಸೇರ್ಪಡೆ ಮಾಡಿದಾಗ ಘಾತೀಯವಾಗಿ ಕಡಿಮೆಯಾಗುತ್ತದೆ.

ಕಾಲ ಕಳೆದಂತೆ ಹಾರ್ಡ್ ವೇರ್ ವೇಗ ಹೆಚ್ಚಾಗುವುದಕ್ಕೆ ಮತ್ತು ನೋಡ್ ಗಳ ನಡೆಯುವಲ್ಲಿ ವ್ಯತ್ಯಾಸಗೊಳ್ಳುವ ಆಸಕ್ತಿಗೆ ಪರಿಹಾರವಾಗಿ, ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಕಠಿಣತೆಯನ್ನು ಪ್ರತಿ ಗಂಟೆಗೆ ಸರಾಸರಿ ಸಂಖ್ಯೆಯ ಬ್ಲಾಕ್ ಗಳನ್ನು ಗುರಿಯಾಗಿಸಿ ಚಲಿಸುವ ಸರಾಸರಿಯನ್ನು ನಿರ್ಧರಿಸಲಾಗುತ್ತದೆ. ಅವು ಅತ್ಯಂತ ವೇಗವಾಗಿ ಉತ್ಪಾದನೆಗೊಂಡರೆ, ಕಠಿಣತೆ ಹೆಚ್ಚಾಗುತ್ತದೆ.

5. ನೆಟ್ವರ್ಕ್

ನೆಟ್ವರ್ಕ್ ನಡೆಸಲು ಹಂತಗಳು ಈ ಕೆಳಕಂಡಂತಿವೆ:

1) ಹೊಸ ವಹಿವಾಟುಗಳು ಎಲ್ಲ ನೋಡ್ ಗಳಲ್ಲಿ ಪ್ರಸಾರವಾಗುತ್ತವೆ
2) ಪ್ರತಿ ನೋಡ್ ಕೂಡಾ ಹೊಸ ವಹಿವಾಟುಗಳನ್ನು ಬ್ಲಾಕ್ ನೊಳಕ್ಕೆ ಸಂಗ್ರಹಿಸುತ್ತದೆ.
3) ಪ್ರತಿ ನೋಡ್ ತನ್ನ ಬ್ಲಾಕ್ ಗೆ ಕಠಿಣ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಅನ್ನು ಕಂಡುಕೊಳ್ಳುವಲ್ಲಿ ಕೆಲಸ ಮಾಡುತ್ತದೆ.
4) ನೋಡ್ ಗಳು ವಹಿವಾಟುಗಳು ಮೌಲಿಕವಾಗಿದ್ದರೆ ಮತ್ತು ಈಗಾಗಲೇ ವ್ಯಯಿಸದೇ ಇದ್ದಲ್ಲಿ ಮಾತ್ರ ಬ್ಲಾಕ್ ಅನ್ನು ಅನುಮೋದಿಸುತ್ತವೆ.
5) ನೋಡ್ ಗಳು ಬ್ಲಾಕ್ ನಲ್ಲಿ ಅವುಗಳ ಅನುಮೋದನೆಯನ್ನು ಹಿಂದಿನ ಹ್ಯಾಶ್ ನಂತೆ ಅನುಮೋದನೆಯಾದ ಬ್ಲಾಕ್ ನ ಹ್ಯಾಶ್ ಬಳಸಿ ಸರಪಳಿಯಲ್ಲಿ ಮುಂದಿನ ಬ್ಲಾಕ್ ಸೃಷ್ಟಿಸುವತ್ತ ಕೆಲಸ ಮಾಡಿ ಒಪ್ಪುತ್ತವೆ.
6) ನೋಡ್ ಗಳನ್ನು ಸರಿಯಾದ ಅತ್ಯಂತ ದೊಡ್ಡ ಸರಪಳಿ ಎಂದು ಸದಾ ಪರಿಗಣಿಸಲಾಗುತ್ತದೆ ಮತ್ತು ಅದನ್ನು ವಿಸ್ತರಿಸಲೂ ಕೆಲಸ ಮಾಡಲಾಗುತ್ತದೆ.

ಎರಡು ನೋಡ್ ಗಳು ಏಕಕಾಲಕ್ಕೆ ಮುಂದಿನ ಬ್ಲಾಕ್ ವಿಭಿನ್ನ ಆವೃತ್ತಿಗಳನ್ನು ಪ್ರಸಾರ ಮಾಡಿದರೆ ಕೆಲ ನೋಡ್ ಗಳು ಒಂದು ಅಥವಾ ಇನ್ನೊಂದನ್ನು ಮೊದಲು ಪಡೆಯಬಹುದು. ಅಂತಹ ಸಂದರ್ಭದಲ್ಲಿ ಅವು ಮೊದಲು ಸ್ವೀಕರಿಸಿದ ನೋಡ್ ಮೇಲೆ ಕೆಲಸ ಮಾಡುತ್ತವೆ, ಆದರೆ ಅದು ಉದ್ದವಾದರೆ ಇತರೆ ಶಾಖೆಯನ್ನು ಉಳಿಸುತ್ತದೆ. ಮುಂದಿನ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಕಂಡುಬಂದಾಗ ಈ ಬಂಧನ ಮುರಿಯುತ್ತದೆ ಮತ್ತು ಒಂದು ಶಾಖೆಯು ಉದ್ದವಾಗುತ್ತದೆ, ಇನ್ನೊಂದು ಶಾಖೆಯ ಮೇಲೆ ಕೆಲಸ ಮಾಡುವ ನೋಡ್ ಗಳು ನಂತರ ಉದ್ದವಾದುದಕ್ಕೆ ಬದಲಾಗುತ್ತವೆ.

ಪ್ರಸಾರವಾಗುವ ಹೊಸ ವಹಿವಾಟು ಎಲ್ಲ ನೋಡ್ ಗಳನ್ನೂ ಅಗತ್ಯವಾಗಿ ತಲುಪಬೇಕಿಲ್ಲ. ಅವು ಹಲವು ನೋಡ್ ಗಳನ್ನು ತಲುಪುವವರೆಗೂ ಉದ್ದಕ್ಕೆ ಮುನ್ನ ಅವು ಬ್ಲಾಕ್ ಗೆ ಸೇರುತ್ತವೆ. ಬ್ಲಾಕ್ ಪ್ರಸಾರಗಳು ಬರಲಾದ ಸಂದೇಶಗಳ ಕುರಿತು ತಾಳ್ಮೆಯಿಂದಲೂ ಇರುತ್ತವೆ. ನೋಡ್ ಬ್ಲಾಕ್ ಅನ್ನು ಸ್ವೀಕರಿಸದಿದ್ದರೆ ಅದು ಮುಂದಿನ ಬ್ಲಾಕ್ ಸ್ವೀಕರಿಸಿದಾಗ ಕೋರುತ್ತದೆ ಮತ್ತು ತಪ್ಪಿಹೋದುದನ್ನು ಅರಿಯುತ್ತದೆ.

6. ಪ್ರೋತ್ಸಾಹಕ

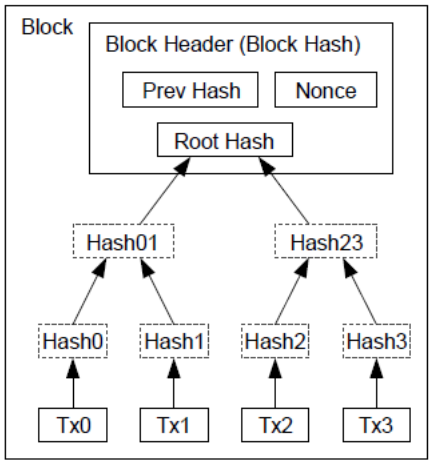
ಸಾಂಪ್ರದಾಯಿಕವಾಗಿ ಬ್ಲಾಕ್ ನಲ್ಲಿನ ಮೊದಲ ವಹಿವಾಟು ವಿಶೇಷ ವಹಿವಾಟು ಆಗಿದ್ದು ಅದು ಬ್ಲಾಕ್ ಸೃಷ್ಟಿಸಿದ ಕ್ರಿಯೇಟರ್ ಮಾಲೀಕತ್ವದ ಹೊಸ ಕಾಯಿನ್ ನಿಂದ ಪ್ರಾರಂಭವಾಗುತ್ತದೆ. ಇದು ನೋಡ್ ಗಳಿಗೆ ನೆಟ್ವರ್ಕ್ ಬೆಂಬಲಿಸಲು ಪ್ರೋತ್ಸಾಹಕ ಸೇರಿಸುತ್ತದೆ, ಮತ್ತು ಅವುಗಳನ್ನು ವಿತರಿಸಲು ಯಾವುದೇ ಕೇಂದ್ರ ಸಂಸ್ಥೆ ಇಲ್ಲದ ಕಾರಣ ಪ್ರಾರಂಭದಲ್ಲಿ ಕಾಯಿನ್ ಗಳನ್ನು ಪ್ರಸರಣಕ್ಕೆ ವಿತರಿಸುವ ಮಾರ್ಗ ಒದಗಿಸುತ್ತದೆ. ಹೊಸ ಕಾಯಿನ್ ಗಳ ಸತತ ಸ್ಥಿರವಾದ ಸೇರ್ಪಡೆಯು ಚಿನ್ನದ ಗಣಿಗಾರರ ರೀತಿಯಲ್ಲಿ ಸಂಪನ್ಮೂಲಗಳನ್ನು ಖರ್ಚು ಮಾಡುವ ಮೂಲಕ ಚಿನ್ನವನ್ನು ಪ್ರಸರಣಕ್ಕೆ ತರುವಂತೆಯೇ ಇರುತ್ತದೆ. ನಮ್ಮ ಪ್ರಕರಣದಲ್ಲಿ ಸಿಪಿಯು ಸಮಯ ಮತ್ತು ವಿದ್ಯುಚ್ಛಕ್ತಿಯನ್ನು ವ್ಯಯಿಸಲಾಗುತ್ತದೆ.

ಈ ಪ್ರೋತ್ಸಾಹಕವನ್ನು ವಹಿವಾಟಿನ ಶುಲ್ಕ ಎಂದೂ ವಿಧಿಸಬಹುದು. ವಹಿವಾಟಿನ ಔಟ್ ಪುಟ್ ಮೌಲ್ಯವು ಇನ್ ಪುಟ್ ಮೌಲ್ಯಕ್ಕಿಂತ ಕಡಿಮೆ ಇದ್ದಲ್ಲಿ ವಹಿವಾಟಿನ ವ್ಯತ್ಯಾಸ ಶುಲ್ಕವನ್ನು ವಹಿವಾಟು ಒಳಗೊಂಡಿರುವ ಬ್ಲಾಕ್ ಮೌಲ್ಯಕ್ಕೆ ಪ್ರೋತ್ಸಾಹಕ ಮೌಲ್ಯಕ್ಕೆ ಸೇರ್ಪಡೆ ಮಾಡಲಾಗುತ್ತದೆ. ಪೂರ್ವ ನಿಗದಿತ ಕಾಯಿನ್ ಗಳ ಸಂಖ್ಯೆಯ ಕಾಯಿನ್ ಗಳನ್ನು ಪ್ರಸರಣಕ್ಕೆ ಪ್ರವೇಶಿಸಿದ ನಂತರ ಈ ಪ್ರೋತ್ಸಾಹಕವನ್ನು ಇಡಿಯಾಗಿ ವಹಿವಾಟಿನ ಶುಲ್ಕವಾಗಿ ಪರಿವರ್ತಿಸಬಹುದು ಮತ್ತು ಸಂಪೂರ್ಣ ಹಣದುಬ್ಬರ ಮುಕ್ತವಾಗಿಸಬಹುದು.

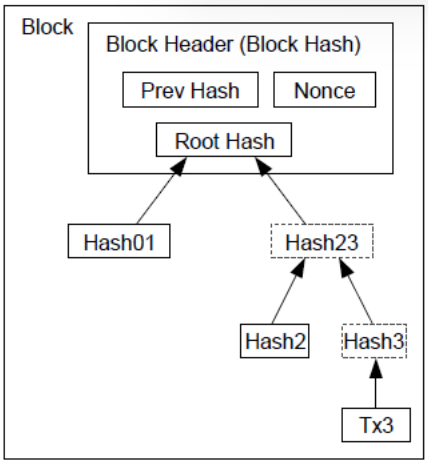
ಈ ಪ್ರೋತ್ಸಾಹಕವು ನೋಡ್ ಗಳು ಪ್ರಾಮಾಣಿಕವಾಗಿರುವಂತೆ ಉತ್ತೇಜಿಸಲು ನೆರವಾಗಬಹುದು. ದುರಾಸೆಯ ದಾಳಿಕೋರ ಎಲ್ಲ ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳಿಗಿಂತ ಹೆಚ್ಚು ಸಿಪಿಯು ಶಕ್ತಿಯನ್ನು ಜೋಡಿಸಬಲ್ಲವನಾದರೆ ಆತ ಅದನ್ನು ತನ್ನ ಪಾವತಿಗಳನ್ನು ಜನರಿಂದ ಕದಿಯುವ ಮೂಲಕ ವಂಚಿಸಲು ಬಳಸಬಹುದು ಅಥವಾ ಅದನ್ನು ಹೊಸ ಕಾಯಿನ್ ಗಳ ಉತ್ಪಾದಿಸಲು ಬಳಸಬಹುದು. ಆತ ಹೆಚ್ಚಾಗಿ ನಿಯಮಗಳ ಮೂಲಕ ಆಡುವುದೇ ಹೆಚ್ಚು ಲಾಭದಾಯಕ ಎಂದು ಕಂಡುಕೊಳ್ಳಬಹುದು, ಅಂತಹ ನಿಯಮಗಳು ಪ್ರತಿಯೊಬ್ಬರದ್ದೂ ಒಗ್ಗೂಡಿಸಿದಾಗಲೂ ವ್ಯವಸ್ಥೆಯನ್ನು ಮತ್ತು ಆತನದೇ ಸಂಪತ್ತಿನ ಮಾನ್ಯತೆಯನ್ನು ದುರ್ಬಲಗೊಳಿಸುವುದರ ಬದಲಾಗಿ ಹೆಚ್ಚು ಹೊಸ ಕಾಯಿನ್ ಗಳ ಮೂಲಕ ಆತನಿಗೆ ಅನುಕೂಲ ಕಲ್ಪಿಸಬಹುದು.

7. ಡಿಸ್ಕ್ ಸ್ಥಳದ ಮರು ಪಡೆದುಕೊಳ್ಳುವಿಕೆ

ಕಾಯಿನ್ ನಲ್ಲಿ ಹೊಚ್ಚಹೊಸ ವಹಿವಾಟನ್ನು ತಕ್ಕಷ್ಟು ಬ್ಲಾಕ್ ಗಳ ಅಡಿಯಲ್ಲಿ ಮುಚ್ಚಿದ ನಂತರ ಡಿಸ್ಕ್ ಸ್ಥಳವನ್ನು ಉಳಿಸಲು ವ್ಯಯಿಸಿದ ವಹಿವಾಟುಗಳನ್ನು ನಿರ್ಲಕ್ಷಿಸಬಹುದು. ಬ್ಲಾಕ್ ಹ್ಯಾಶ್ ಮುರಿಯದೆ ಇದನ್ನು ಸಾಧ್ಯವಾಗಿಸಲು ವಹಿವಾಟುಗಳನ್ನು ಮರ್ಕೆಲ್ ಟ್ರೀಯಲ್ಲಿ [7][2][5], ಹ್ಯಾಶ್ ಮಾಡಬೇಕು, ಅದರಲ್ಲಿ ರೂಟ್ ಅನ್ನು ಬ್ಲಾಕ್ ನ ಹ್ಯಾಶ್ ನಲ್ಲಿ ಒಳಗೊಳ್ಳಬೇಕು. ಹಳೆಯ ಬ್ಲಾಕ್ ಗಳನ್ನು ನಂತರ ಮರದ ಶಾಖೆಗಳಿಂದ ಮೊಂಡು ಮಾಡುವ ಮೂಲಕ ಕಿರಿದಾಗಿಸಬಹುದು. ಒಳಾಂಗಣ ಹ್ಯಾಶ್ ಸಂಗ್ರಹಿಸಬೇಕು.



Transactions Hashed in a Merkle Tree

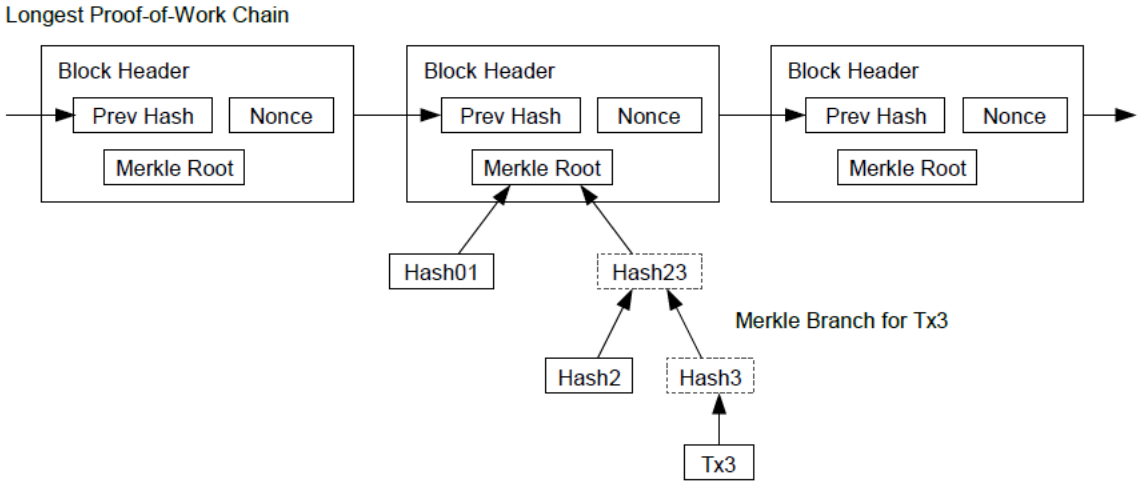


After Pruning Tx0-2 from the Block

ವಹಿವಾಟುಗಳಿಲ್ಲದ ಬ್ಲಾಕ್ ಹೆಡರ್ 80 ಬೈಟ್ ಗಳಿರಬಹುದು. ಈ ಬ್ಲಾಕ್ ಗಳನ್ನು ಪ್ರತಿ 10 ನಿಮಿಷಗಳಿಗೆ ಉತ್ಪಾದಿಸಿದರೆ 80 ಬೈಟ್ಸ್ ಪ್ರತಿ ವರ್ಷಕ್ಕೆ * 6 * 24 * 365 = 4.2 MB. ಕಂಪ್ಯೂಟರ್ ಸಿಸ್ಟಂಗಳು ಸಾಮಾನ್ಯವಾಗಿ 2008ರಲ್ಲಿ 2ಜಿಬಿ RAM ನೊಂದಿಗೆ ಮಾರಾಟವಾಗುತ್ತಿವೆ ಮತ್ತು ಮೂರ್ಸ್ ನಿಯಮವು ಪ್ರಸ್ತುತ ಪ್ರಗತಿಯನ್ನು ವರ್ಷಕ್ಕೆ 1.2 ಜಿಬಿ ಎಂದು ಊಹಿಸುತ್ತಿದ್ದು ಸಂಗ್ರಹವು ಬ್ಲಾಕ್ ಹೆಡರ್ ಗಳನ್ನು ಮೆಮೊರಿಯಲ್ಲಿ ಇರಿಸಿದರೂ ಸಂಗ್ರಹ ಸಮಸ್ಯೆಯಲ್ಲ.

8. ಸರಳೀಕೃತ ಪಾವತಿಯ ಪರಿಶೀಲನೆ

ಪಾವತಿಗಳನ್ನು ಪೂರ್ಣ ನೆಟ್ವರ್ಕ್ ನೋಡ್ ಚಾಲಿಸದೆ ಪರಿಶೀಲಿಸುವುದು ಸಾಧ್ಯ. ಬಳಕೆದಾರ ಬ್ಲಾಕ್ ಹೆಡರ್ ಗಳ ಪ್ರತಿಯನ್ನು ಇರಿಸಿಕೊಳ್ಳಬೇಕು, ಅದರಿಂದ ಆತ ನೆಟ್ವರ್ಕ್ ನೋಡ್ ಅನ್ನು ವಿಚಾರಿಸಿ ತನ್ನಲ್ಲಿ ಅತ್ಯಂತ ದೊಡ್ಡ ಸರಪಳಿ ಇದೆ ಎಂದು ಮನವರಿಕೆಯಾಗುವವರೆಗೆ ಮತ್ತು ಟೈಂಸ್ಟಾಂಪ್ ಮಾಡಿದ ಬ್ಲಾಕ್ ಗೆ ವಹಿವಾಟನ್ನು ಸಂಪರ್ಕಿಸಿ ಮರ್ಕೆಲ್ ಶಾಖೆ ಪಡೆಯಬಹುದು. ಆತ ತನಗೆ ಸ್ವತಃ ವಹಿವಾಟನ್ನು ಪರೀಕ್ಷಿಸಲು ಸಾಧ್ಯವಿಲ್ಲ, ಬದಲಿಗೆ ಅದನ್ನು ಸರಪಳಿಯ ಸ್ಥಳಕ್ಕೆ ಸಂಪರ್ಕಿಸಬಹುದು, ಆತ ನೆಟ್ವರ್ಕ್ ನೋಡ್ ಅದನ್ನು ಅನುಮೋದಿಸಿದೆ ಎಂದು ಮತ್ತು ನಂತರ ನೆಟ್ವರ್ಕ್ ಅದನ್ನು ಅನುಮೋದಿಸಿದೆ ಎಂದು ಮತ್ತಷ್ಟು ದೃಢೀಕರಿಸಲು ಸೇರಿಸಿದ ಬ್ಲಾಕ್ ಗಳನ್ನು ಕಾಣಬಹುದು.

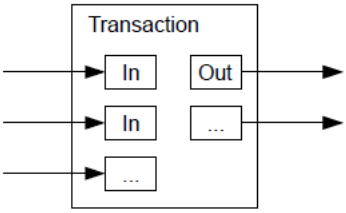


ಹಾಗೆ ಪರಿಶೀಲನೆಯು ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳು ನೆಟ್ವರ್ಕ್ ಅನ್ನು ನಿಯಂತ್ರಿಸುವವರೆಗೆ ವಿಶ್ವಾಸಾರ್ಹವಾಗಿರುತ್ತದೆ, ಆದರೆ ದಾಳಿಕೋರನಿಂದ ನೆಟ್ವರ್ಕ್ ಪ್ರಭಾವಿತವಾದರೆ ಹೆಚ್ಚು ದಾಳಿಗೆ ಈಡಾಗುತ್ತದೆ. ನೆಟ್ವರ್ಕ್ ನೋಡ್ ಗಳು ಸ್ವತಃ ವಹಿವಾಟುಗಳನ್ನು ಪರಿಶೀಲಿಸಬಹುದಾದರೆ ಸರಳೀಕೃತ ವಿಧಾನವನ್ನು ದಾಳಿಕೋರನ ಫ್ಯಾಬ್ರಿಕೇಟೆಡ್ ವಹಿವಾಟುಗಳು ದಾಳಿಕೋರ ನೆಟ್ವರ್ಕ್ ಅನ್ನು ಮೀರಲು ಪ್ರಯತ್ನಿಸುವವರೆಗೂ ಸರಳೀಕರಿಸಿದ ವಿಧಾನವನ್ನು ದಾರಿ ತಪ್ಪಿಸಬಹುದು. ಇದರ ವಿರುದ್ಧ ರಕ್ಷಣೆಗೆ ಒಂದು ಕಾರ್ಯತಂತ್ರವೆಂದರೆ ಇದು ಮೌಲಿಕವಲ್ಲದ ಬ್ಲಾಕ್ ಪತ್ತೆ ಮಾಡಿದಾಗ ನೆಟ್ವರ್ಕ್ ನೋಡ್ ಗಳಿಂದ ಎಚ್ಚರಿಕೆಯನ್ನು ಒಪ್ಪಿಕೊಳ್ಳಬಹುದು, ಇದರಿಂದ ಬಳಕೆದಾರನ ಸಾಫ್ಟ್ ವೇರ್ ಪೂರ್ಣ ಬ್ಲಾಕ್ ಡೌನ್ಲೋಡ್ ಮಾಡಬೇಕಾಗುತ್ತದೆ ಮತ್ತು ಅಸಂಗತತೆ ದೃಢೀಕರಿಸಲು ವಹಿವಾಟುಗಳನ್ನು ಎಚ್ಚರಿಸುತ್ತದೆ. ಆಗಾಗ್ಗೆ ಪಾವತಿಗಳನ್ನು ಪಡೆಯುವ ವ್ಯಾಪಾರವು ಹೆಚ್ಚು ಸ್ವತಂತ್ರ ಭದ್ರತೆ ಮತ್ತು ತ್ವರಿತ ಪರಿಶೀಲನೆಗೆ ತಮ್ಮದೇ ಆದ ನೋಡ್ ಗಳನ್ನು ಚಾಲಿಸಲು ಬಯಸಬಹುದು.

9. ಒಗ್ಗೂಡಿಕೆ ಮತ್ತು ವಿಭಜನೆಯ ಮೌಲ್ಯ

ಕಾಯಿನ್ ಗಳನ್ನು ವೈಯಕ್ತಿಕವಾಗಿ ನಿರ್ವಹಿಸುವುದು ಸಾಧ್ಯವಿದ್ದರೂ ವರ್ಗಾವಣೆಯಲ್ಲಿ ಪ್ರತಿ ಸೆಂಟ್ ಗೆ ಪ್ರತ್ಯೇಕ ವಹಿವಾಟು ಮಾಡುವುದು ಒರಟಾಗಿರುತ್ತದೆ. ಮೌಲ್ಯದ ವಿಭಜನೆ ಮತ್ತು ಒಗ್ಗೂಡಿಕೆ ಸಾಧ್ಯವಾಗಿಸಲು ವಹಿವಾಟುಗಳು ಹಲವು ಇನ್ ಪುಟ್ ಮತ್ತು ಔಟ್ ಪುಟ್ ಗಳನ್ನು ಹೊಂದಿರುತ್ತವೆ. ಸಾಮಾನ್ಯವಾಗಿ ದೊಡ್ಡ ಹಿಂದಿನ ವಹಿವಾಟಿನಲ್ಲಿ ಒಂದು ಇನ್ಪುಟ್ ಇರುತ್ತದೆ ಅಥವಾ ಸಣ್ಣ ಮೊತ್ತಗಳನ್ನು ಒಗ್ಗೂಡಿಸುವ ಹಲವು ಇನ್ ಪುಟ್ ಗಳಿರುತ್ತವೆ ಮತ್ತು ಹೆಚ್ಚೆಂದರೆ

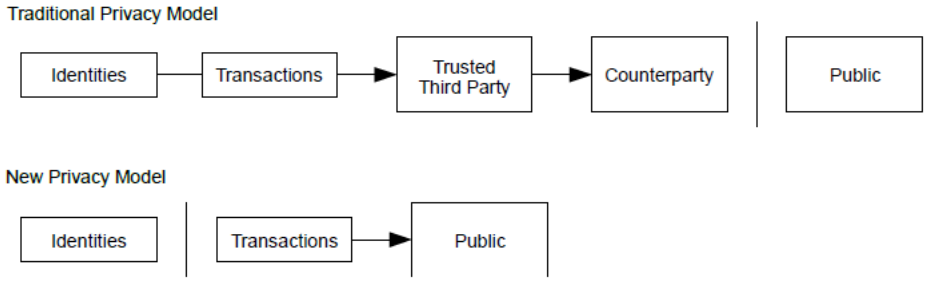
ಎರಡು ಇನ್ ಪುಟ್ ಗಳಿರುತ್ತವೆ: ಒಂದು ಪಾವತಿಗೆ ಮತ್ತು ಇನ್ನೊಂದು ಕಳುಹಿಸಿದವನಿಗೆ ಚಿಲ್ಲರೆ ಹಿಂದಿರುಗಿಸಲು.



ಇಲ್ಲಿ ಗಮನಿಸಬೇಕಾದುದು ವಹಿವಾಟು ಹವಾರು ವಹಿವಾಟುಗಳನ್ನು ಆಧರಿಸಿರುತ್ತದೆ ಮತ್ತು ಆ ವಹಿವಾಟುಗಳು ಹಲವು ಹೆಚ್ಚಿನದಕ್ಕೆ ಆಧರಿಸಿರುತ್ತವೆ ಎನ್ನುವುದು ಇಲ್ಲಿ ಸಮಸ್ಯೆಯಲ್ಲ. ವಹಿವಾಟಿನ ಇತಿಹಾಸದ ಸಂಪೂರ್ಣ ಪ್ರತ್ಯೇಕ ಪ್ರತಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳು ಅಗತ್ಯವೇ ಇರುವುದಿಲ್ಲ.

10. ಖಾಸಗಿತನ

ಸಾಂಪ್ರದಾಯಿಕ ಬ್ಯಾಂಕಿಂಗ್ ಮಾದರಿಯು ಭಾಗವಹಿಸಿರುವ ವ್ಯಕ್ತಿಗಳ ಮತ್ತು ಮೂರನೇ ವ್ಯಕ್ತಿಯ ನಡುವೆ ಮಾಹಿತಿಯ ಲಭ್ಯತೆಯನ್ನು ಸೀಮಿತಗೊಳಿಸಿ ಖಾಸಗಿತನದ ಮಟ್ಟವನ್ನು ಸಾಧಿಸುತ್ತದೆ. ಎಲ್ಲ ವಹಿವಾಟುಗಳನ್ನೂ ಸಾರ್ವಜನಿಕವಾಗಿ ಪ್ರಕಟಿಸುವ ಅಗತ್ಯ ಈ ವಿಧಾನವನ್ನು ತಡೆಯುತ್ತದೆ, ಆದರೆ ಖಾಸಗಿತನವನ್ನು ಮತ್ತೊಂದು ಸ್ಥಳಕ್ಕೆ ಸಾರ್ವಜನಿಕ ಕೀಗಳನ್ನು ಅನಾಮಿಕವಾಗಿರಿಸಿ. ಮಾಹಿತಿಯ ಹರಿವು ತಡೆಯುವ ಮೂಲಕ ಇನ್ನೂ ಕಾಪಾಡಬಹುದು. ಸಾರ್ವಜನಿಕರು ವಹಿವಾಟನ್ನು ಯಾರೊಂದಿಗೂ ಸಂಪರ್ಕ ನೀಡದೆ ಯಾರೋ ಒಂದು ಮೊತ್ತವನ್ನು ಮತ್ತೊಬ್ಬರಿಗೆ ಕಳುಹಿಸಿದ್ದಾರೆ ಎಂದು ಕಾಣಬಹುದು. ಇದು ಷೇರು ವಿನಿಮಯ ಕೇಂದ್ರಗಳಲ್ಲಿ ಬಿಡುಗಡೆ ಮಾಡುವ ಮಾಹಿತಿಯ ಮಟ್ಟದಂತೆಯೇ ಇರುತ್ತದೆ, ಅಲ್ಲಿ ವೈಯಕ್ತಿಕ ವಹಿವಾಟುಗಳ ಸಮಯ ಮತ್ತು ಗಾತ್ರದ "ಟೇಪ್" ಅನ್ನು ಸಾರ್ವಜನಿಕಗೊಳಿಸಲಾಗುತ್ತದೆ, ಆದರೆ ವ್ಯಕ್ತಿಗಳು ಯಾರು ಎಂದು ಹೇಳುವುದಿಲ್ಲ.



ಹೆಚ್ಚುವರಿ ಫೈರ್ ವಾಲ್ ಆಗಿ ಪ್ರತಿ ವಹಿವಾಟಿಗೂ ಹೊಸ ಕೀ ಜೋಡಿಯನ್ನು ಬಳಸುವ ಮೂಲಕ ಅವುಗಳನ್ನು ಸಾಮಾನ್ಯ ಮಾಲೀಕನಿಗೆ ಸಂಪರ್ಕಿಸಬಹುದು. ಹಲವು ಇನ್ಪುಟ್ ವಹಿವಾಟುಗಳಿಂದ ಸ್ವಲ್ಪ ಲಿಂಕಿಂಗ್ ಅನ್ನು ತಪ್ಪಿಸಲು ಸಾಧ್ಯವೇ ಇಲ್ಲ, ಅದು ಅವರ ಇನ್ ಪುಟ್ ಗಳು ಅದೇ ಮಾಲೀಕನ ಮಾಲೀಕತ್ವ ಹೊಂದಿವೆ ಎಂದು ಅನಾವರಣಗೊಳಿಸಬೇಕು. ಇಲ್ಲಿ ರಿಸ್ಕ್ ಎಂದರೆ ಕೀ ಮಾಲೀಕ ಅನಾವರಣಗೊಂಡರೆ ಲಿಂಕಿಂಗ್ ಅದೇ ಮಾಲೀಕನಿಗೆ ಸಂಬಂಧಿಸಿದ ಇತರೆ ವಹಿವಾಟುಗಳನ್ನು ಅನಾವರಣಗೊಳಿಸಬಹುದು.

11. ಲೆಕ್ಕಾಚಾರಗಳು

ದಾಳಿಕೋರ ಪ್ರಾಮಾಣಿಕ ಸರಪಳಿಗಿಂತ ವೇಗವಾಗಿ ಪರ್ಯಾಯ ಸರಪಳಿಯನ್ನು ಉತ್ಪಾದಿಸಲು ಪ್ರಯತ್ನಿಸಿದ ಸನ್ನಿವೇಶ ಪರಿಗಣಿಸೋಣ. ಇದನ್ನು ಒಪ್ಪಿಕೊಂಡರೂ ಅದು ಅನಿಯಂತ್ರಿತ ಬದಲಾವಣೆಗಳಿಗೆ ವ್ಯವಸ್ಥೆಯನ್ನು ತೆರೆಯುವುದಿಲ್ಲ, ಅಂದರೆ ಅಗೋಚರವಾದುದರಿಂದ ಮೌಲ್ಯ ಸೃಷ್ಟಿಸುವುದು ಅಥವಾ ದಾಳಿಕೋರನಿಗೆ ಎಂದೂ ಸೇರದ ಹಣವನ್ನು ತೆಗೆದುಕೊಳ್ಳುವುದು ಸಾಧ್ಯವಿಲ್ಲ. ನೋಡ್ ಗಳು ಮೌಲಿಕವಲ್ಲದ ವಹಿವಾಟುಗಳನ್ನು ಪಾವತಿ ಎಂದು ಒಪ್ಪಿಕೊಳ್ಳುವುದಿಲ್ಲ, ಮತ್ತು ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳು ತಮ್ಮನ್ನು ನಿಯಂತ್ರಿಸುವ ಬ್ಲಾಕ್ ಅನ್ನು ಎಂದಿಗೂ ಒಪ್ಪಿಕೊಳ್ಳುವುದಿಲ್ಲ. ದಾಳಿಕೋರ ತಾನು ಇತ್ತೀಚೆಗೆ ವ್ಯಯಿಸಿದ ಹಣವನ್ನು ಹಿಂಪಡೆಯಲು ತನ್ನದೇ ವಹಿವಾಟುಗಳಲ್ಲಿ ಒಂದನ್ನು ಬದಲಾಯಿಸಲು ಮಾತ್ರ ಪ್ರಯತ್ನಿಸಬಹುದು.

ಪ್ರಾಮಾಣಿಕ ಸರಪಳಿ ಮತ್ತು ದಾಳಿಕೋರನ ಸರಪಳಿಯನ್ನು ಬಿನೋಮಿಯಲ್ ರಾಂಡಂ ವಾಕ್ ಎಂದು ಗುರುತಿಸಬಹುದು. ಈ ಯಶಸ್ಸಿನ ಕಾರ್ಯಕ್ರಮವು ಪ್ರಾಮಾಣಿಕ ಸರಪಳಿಯು ಒಂದು ಬ್ಲಾಕ್ ಮೂಲಕ ವಿಸ್ತರಿಸುವ ಮೂಲಕ ತನ್ನ ಮುನ್ನಡೆಯನ್ನು +1ಕ್ಕೆ ಹೆಚ್ಚಿಸುತ್ತದೆ ಮತ್ತು ವಿಫಲ ಕಾರ್ಯಕ್ರಮವು ದಾಳಿಕೋರನ ಸರಪಳಿಯನ್ನು

ಒಂದು ಬ್ಲಾಕ್ ನಿಂದ ವಿಸ್ತರಿಸುತ್ತದೆ, ಇದರಿಂದ ಅಂತರವನ್ನು -1ಕ್ಕೆ ಕಡಿಮೆ ಮಾಡುತ್ತದೆ.

ದಾಳಿಕೋರನ ಸಂಭವನೀಯತೆಯು ನೀಡಲಾದ ಕೊರತೆಯಿಂದ ಹಿಡಿದುಕೊಳ್ಳುವುದು ಗ್ಯಾಂಬ್ಲರ್ಸ್ ರುಯಿನ್ ಸಮಸ್ಯೆಗೆ ಹೋಲುತ್ತದೆ. ಜೂಜುಕೋರನು ಅನಿಯಮಿತ ಕ್ರೆಡಿಟ್ ನೊಂದಿಗೆ ಕೊರತೆಯಿಂದ ಪ್ರಾರಂಭಿಸಿದರೆ ಮತ್ತು ಬ್ರೇಕ್ ಈವನ್ ತಲುಪಲು ಅನಂತ ಸಂಖ್ಯೆಯ ಪ್ರಯತ್ನಗಳೊಂದಿಗೆ ಆಡಿದಾಗ. ನಾವು ಆತ ಎಂದಿಗೂ ಬ್ರೇಕ್ ಈವನ್ ಸಂಭವನೀಯತೆ ತಲುಪುವುದಿಲ್ಲ ಎಂದು ಲೆಕ್ಕ ಹಾಕಬಹುದು ಅಥವಾ ದಾಳಿಕೋರ ಪ್ರಾಮಾಣಿಕ ಸರಪಳಿ ಪಡೆದುಕೊಳ್ಳಲು ಸಾಧ್ಯವೇ ಇಲ್ಲ, ಅದು ಹೀಗಿರುತ್ತದೆ [8]:

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

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

$p > q$ ಎಂದು ಭಾವಿಸಿದಲ್ಲಿ ದಾಳಿಕೋರ ಹಿಡಿದುಕೊಳ್ಳುವ ಬ್ಲಾಕ್ ಗಳ ಸಂಖ್ಯೆ ಅಪಾರವಾಗಿ ಕುಸಿಯುವ ಸಂಭವನೀಯತೆಯು ಹೆಚ್ಚಾಗುತ್ತದೆ. ಆತನ ವಿರುದ್ಧ ಪ್ರತಿರೋಧಗಳಿದ್ದರೂ ಆತ ಮುಂಚೆಯೇ ಮುನ್ನಡೆಯದಿದ್ದರೆ ಆತ ಮತ್ತಷ್ಟು ಹಿಂದಕ್ಕೆ ಕುಸಿಯುವುದರಿಂದ ಆತನ ಸಾಧ್ಯತೆಗಳು ಅತ್ಯಂತ ಕಡಿಮೆಯಾಗಿರುತ್ತವೆ.

ನಾವು ಈಗ ಎಷ್ಟು ಕಾಲ ಹೊಸ ವಹಿವಾಟು ಸ್ವೀಕರಿಸುವವನು ಕಳುಹಿಸಿದವನು ವಹಿವಾಟನ್ನು ಬದಲಾಯಿಸುವುದಿಲ್ಲ ಎಂದು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಲು ಎಷ್ಟು ಕಾಲ ಕಾಯಬೇಕು ಎನ್ನುವುದನ್ನು ಪರಿಗಣಿಸಬೇಕು. ಕಳುಹಿಸಿದವನು ದಾಳಿಕೋರ ಎಂದು ಅಂದುಕೊಳ್ಳೋಣ, ಆತನಿಗೆ ಸ್ವೀಕರಿಸುವವನಿಗೆ ಪಾವತಿ ಮಾಡಿರುವುದನ್ನು ಸ್ವಲ್ಪ ಕಾಲ ನಂಬುವಂತೆ ಮಾಡಬೇಕು, ನಂತರ ಸ್ವಲ್ಪ ಕಾಲ ಕಳೆದಂತೆ ಅದನ್ನು ತನಗೆ ಹಿಂದಿರುಗಿಸಿಕೊಳ್ಳಬೇಕು. ಹೀಗೆ ಆದಲ್ಲಿ ಸ್ವೀಕರಿಸಿದವನು ಎಚ್ಚರಿಕೆಯನ್ನು ಪಡಯುತ್ತಾನೆ, ಆದರೆ ಕಳುಹಿಸಿದವನು ಅದು ಬಹಳ ತಡವಾಗಿಲ್ಲ ಎಂದು ಭರವಸೆ ಹೊಂದಿರಬೇಕು.

ಸ್ವೀಕರಿಸಿದವನು ಹೊಸ ಕೀ ಜೋಡಿ ಸೃಷ್ಟಿಸುತ್ತಾನೆ ಮತ್ತು ಸೈನಿಂಗ್ ಮಾಡುವ ಮುನ್ನ ಸಾರ್ವಜನಿಕ ಕೀಯನ್ನು ಕಳುಹಿಸಿದವನಿಗೆ ನೀಡುತ್ತಾನೆ. ಇದು ಕಳುಹಿಸಿದವನಿಗೆ ಅದಕ್ಕೆ ಸತತವಾಗಿ ಕೆಲಸ ಮಾಡುವ ಮೂಲಕ ಮತ್ತು ವಹಿವಾಟನ್ನು ಅದೇ ಕ್ಷಣದಲ್ಲಿ ಕಾರ್ಯಗತಗೊಳಿಸುವ ಮೂಲಕ ಬಹಳ ಮುಂಚೆಯೇ ಬ್ಲಾಕ್ ಗಳ ಸರಪಳಿಯನ್ನು ಸಿದ್ಧಗೊಳಿಸುವುದರಿಂದ ತಡೆಯುತ್ತದೆ. ವಹಿವಾಟನ್ನು ಒಮ್ಮೆ ಕಳುಹಿಸಿದ ನಂತರ ಅಪ್ರಾಮಾಣಿಕ ಪಾವತಿದಾರ ರಹಸ್ಯವಾಗಿ ಸಮಾನಾಂತರ ಸರಪಳಿಯನ್ನು ಕೆಲಸ ಮಾಡಲು ಪ್ರಾರಂಭಿಸಿ ಆತನ ವಹಿವಾಟಿನ ಪರ್ಯಾಯ ಆವೃತ್ತಿಯನ್ನು ನಿರ್ಬಂಧಿಸಲು ಪ್ರಯತ್ನಿಸುತ್ತಾನೆ.

ಸ್ವೀಕರಿಸಿದವನು ವಹಿವಾಟನ್ನು ಬ್ಲಾಕ್ ಗೆ ಸೇರಿಸುವವರೆಗೆ ಕಾಯುತ್ತಾನೆ ಮತ್ತು z ಬ್ಲಾಕ್ ಗಳನ್ನು ನಂತರ ಸಂಪರ್ಕಿಸಲಾಗುತ್ತದೆ. ಆತನಿಗೆ ದಾಳಿಕೋರ ಮಾಡಿರುವ ಪ್ರಗತಿಯ ನಿರ್ದಿಷ್ಟ ಮೊತ್ತ ಗೊತ್ತಿರುವುದಿಲ್ಲ. ಆದರೆ ಪ್ರಾಮಾಣಿಕ ಬ್ಲಾಕ್ ಗಳು ಪ್ರತಿ ಬ್ಲಾಕ್ ಗೆ ಸರಾಸರಿ ನಿರೀಕ್ಷಿತ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳುತ್ತದೆ ಎಂದು ಅರಿಯುತ್ತಾನೆ, ದಾಳಿಕೋರನ ಸಾಮರ್ಥ್ಯದ ಪ್ರಗತಿಯು ನಿರೀಕ್ಷೆಯ ಮೌಲ್ಯದ ಪಾಯಿಸ್ಸನ್ ಡಿಸ್ಟ್ರಿಬ್ಯೂಷನ್ ಆಗಿರುತ್ತದೆ.

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

ದಾಳಿಕೋರ ಈಗ ಹಿಡಿದುಕೊಳ್ಳಬಲ್ಲ ಸಂಭವನೀಯತೆ ಪಡೆಯಲು ನಾವು ಆತನ ಮಾಡಿದ ಪ್ರಗತಿಯ ಪ್ರತಿ ಮೊತ್ತದ ಪಾಯಿಸ್ಸನ್ ಸಾಂದ್ರತೆಯನ್ನು ಆ ಸಮಯದಲ್ಲಿ ಆತ ಹಿಡಿದುಕೊಳ್ಳಬಲ್ಲ ಸಂಭವನೀಯತೆಯೊಂದಿಗೆ ಗುಣಿಸಬೇಕು:

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

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. ಉಪಸಂಹಾರ

ನಾವು ವಿಶ್ವಾಸದ ಮೇಲೆ ಆಧಾರಪಡದ ಎಲೆಕ್ಟ್ರಾನಿಕ್ ವಹಿವಾಟುಗಳ ವ್ಯವಸ್ಥೆಯನ್ನು ಪ್ರಸ್ತಾವಿಸಿದ್ದೇವೆ. ನಾವು ಡಿಜಿಟಲ್ ಸಿಗ್ನೇಚರ್ ಗಳಿಂದ ತಯಾರಿಸಲಾದ ಸಾಮಾನ್ಯ ಕಾಯಿನ್ ಗಳ ಫ್ರೇಮ್ ವರ್ಕ್ ನಿಂದ ಪ್ರಾರಂಭಿಸಿದ್ದೇವೆ, ಅದು ಮಾಲೀಕತ್ವದ ಸದೃಢ ನಿಯಂತ್ರಣ ನೀಡುತ್ತದೆ, ಆದರೆ ಎರಡು ಬಾರಿ ವ್ಯಯಿಸುವುದನ್ನು ತಪ್ಪಿಸಲು ಮಾರ್ಗವನ್ನು ಕಂಡುಕೊಳ್ಳದೇ ಇದ್ದಲ್ಲಿ ಅಸಂಪೂರ್ಣವಾಗುತ್ತದೆ. ಇದನ್ನು ಪರಿಹರಿಸಲು ನಾವು ಪೀರ್-ಟು-ಪೀರ್ ಜಾಲವನ್ನು ಪ್ರಸ್ತಾವಿಸಿದ್ದು ಅದಕ್ಕೆ ವಹಿವಾಟಿನ ಸಾರ್ವಜನಿಕ ಇತಿಹಾಸವನ್ನು ಬಳಸುವ ಮೂಲಕ ದಾಖಲಿಸಬೇಕು ಅದು ಪ್ರಾಮಾಣಿಕ ನೋಡ್ ಗಳನ್ನು ಸಿಪಿಯುವಿನ ಹೆಚ್ಚಿನ ಶಕ್ತಿ ಬಳಸಿ ದಾಳಿಕೋರ ಬದಲಾಯಿಸುವುದು ಕಂಪ್ಯುಟೇಷನಲಿ ಅಸಾಧ್ಯವಾಗುತ್ತದೆ. ನೆಟ್ವರ್ಕ್ ತನ್ನ ರಚನೆಯಿಲ್ಲದ ಸರಳರತೆಯಲ್ಲಿ ಸದೃಢವಾಗಿರುತ್ತದೆ. ನೋಡ್ ಗಳು ಬಹಳ ಕಡಿಮೆ ಸಹಯೋಗದಲ್ಲಿ ಒಟ್ಟಿಗೆ ಕೆಲಸ ಮಾಡುತ್ತವೆ. ಸಂದೇಶಗಳನ್ನು ಯಾವುದೇ ನಿರ್ದಿಷ್ಟ ಸ್ಥಳಕ್ಕೆ ಸೂಚಿಸಿಲ್ಲದಿದ್ದರಿಂದ ಅವುಗಳನ್ನು ಗುರುತಿಸುವ ಅಗತ್ಯವಿಲ್ಲ ಮತ್ತು ಅತ್ಯುತ್ತಮ ಪ್ರಯತ್ನ ಆಧರಿಸಿ ಪೂರೈಸಲು ಮಾತ್ರ ಸಾಧ್ಯವಾಗುತ್ತದೆ. ನೋಡ್ ಗಳು ಅವು ಹೊರಗಿದ್ದಾಗ ಏನು ನಡೆದಿದೆ ಎನ್ನುವುದಕ್ಕೆ ಪ್ರೂಫ್-ಆಫ್-ವರ್ಕ್ ಅನ್ನು ಸಾಕ್ಷಿಯಾಗಿ ಪಡೆದುಕೊಂಡು ಬಯಸಿದಾಗ ನೆಟ್ವರ್ಕ್ ಅನ್ನು ತ್ಯಜಿಸಬಹುದು ಮತ್ತು ಸೇರ್ಪಡೆಗೊಳ್ಳಬಹುದು. ಅವು ತಮ್ಮ ಸಿಪಿಯು

ಶಕ್ತಿಯೊಂದಿಗೆ ಅಂಗೀಕರಿಸುತ್ತವೆ, ಅವುಗಳ ಮೌಲಿಕ ಬ್ಲಾಕ್ ಗಳನ್ನು ಅನುಮೋದಿಸಿದ್ದನ್ನು ಅವುಗಳ ವಿಸ್ತರಣೆಯ ಮೇಲೆ ಕೆಲಸ ಮಾಡಿ ಮನ್ನಿಸುತ್ತವೆ ಮತ್ತು ಮೌಲಿಕವಲ್ಲದ ಬ್ಲಾಕ್ ಗಳನ್ನು ಅವುಗಳ ಮೇಲೆ ಕೆಲಸ ಮಾಡಲು ನಿರಾಕರಿಸಿ ತಿರಸ್ಕರಿಸುತ್ತವೆ. ಅಗತ್ಯವಾದ ನಿಯಮಗಳು ಮತ್ತು ಪ್ರೋತ್ಸಾಹಕಗನ್ನು ಈ ಅನುಮೋದನೆಯ ವ್ಯವಸ್ಥೆಯನ್ನು ಬಳಸುವ ಮೂಲಕ ಬಲಪಡಿಸಬಹುದು.

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.