

## פונקצית האש בבטיחות מידע (עירבול)

בגדול זו פונקציה שלוקחת נתון (לרוב מספרי) והופכת אותו למספר אחר שממנו קשה מאד להסיק מה היה הנתון.

פונקציה טובה היא כזו ששינוי קטן בנתון המקורי, יוצר שינוי גדול בתוצאה. כמו כן התוצאה כדאי שתהיה באורך זהה עבור כל הנתונים. בנוסף, במצב אופטימלי עבור נתונים שונים (כמעט) תמיד נקבל תוצאה שונה.

דוגמה גרועה: סכום הספרות. למשל, נתון של 1234 יתן 10. למה זה גרוע?  
אם נשנה לנתון 1235 נקבל 11 (ממש קרוב לקודם) - מפר את התנאי הראשון.

פונקציה טובה יותר יכולה לעשות את הדבר הבא:  
לכפול את המקור ב 1234. להעלות את התוצאה בריבוע. מזה לקחת ספרות 4 עד 9 (מימין לשמאל) כמספר נפרד, שוב נעלה בריבוע ומהתוצאה הזו ניקח ספרות 4-8 שוב מימין לשמאל.

הנה מספר דוגמאות הפעלה

<u>תוצאה</u>	<u>מקור</u>
57569	99
83533	100
64178	101

זה עונה על שני התנאים הראשונים.

מה המקסימום של ערכי מקור שיניבו תוצאות שונות (לא תהיינה התנגשויות)?

רמז: הסתכלו על גודל התוצאה.

בעולם האמיתי ישנם אלגוריתמים עבור פונקצית האש עם תוצאות ארוכות בהרבה, למשל MD5 או SHA512  
MD5 יוצר תוצאה של 128 ביטים או 32 ספרות הקסה. התוכנית hash.py מדגימה זאת.

נסו להריץ את התוכנית על קלט טיפה שונה והתבוננו בתוצאות.  
אחד השימושים הידועים של פונקצית עירבול היא בבדיקה של אמיתות  
משלוח הודעות. Message Authentication Code או MAC.

אם קיבלתי הודעה ברשת מפלוני, איך אהיה בטוח שההודעה לא שונתה  
בדרך?

אפשר להשתמש גם בהצפנה סימטרית (אם לשני הצדדים בלבד ישנו  
הקוד המשותף) וגם בא-סימטרית. יותר מקובל בסימטרית.

התהליך הוא כזה:

צד א' ששולח הודעה (m), מבצע עליה האש. מצפין את ההאש עם הקוד  
שלו ויוצר מה שניקרא tag (t) ושולח גם את ההודעה הלא מוצפנת m  
וגם את ההאש המוצפן שלה t.

הצד המקבל מפענח את ההאש המוצפן t (וכך מקבל את ההאש של  
ההודעה המקורית) ועושה אותו האש להודעה הלא מוצפנת m. אם  
התוצאה שווה - זה אומר שההודעה לא שונתה.

תהליך זה נקרא גם: Data Integrity

הערה: אפשר ובדרך כלל גם רצוי להצפין את ההודעה המקורית עם  
אותו מפתח ואז מוסיפים עוד שלב בשליחה ובקבלה. השולח מצפין עם  
המפתח את m שהופכת ל s, והמקבל מפענח את s לפני ההפעלת  
ההאש.