

Horse-MinHash: High-Performance and Secure Jaccard Similarity Estimation for Cloud Storage

Ben Trovato
Institute for Clarity in Documentation
Dublin, Ireland
trovato@corporation.com

Lars Thörvöld
The Thörvöld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France
vb@rocquencourt.com

Jörg von Árbach
University of Tübingen
Tübingen, Germany
jaerbach@uni-tuebingen.edu
myprivate@email.com
second@affiliation.mail

Wang Xiu Ying
Zhe Zuo
East China Normal University
Shanghai, China
firstname.lastname@ecnu.edu.cn

Donald Fauntleroy Duck
Scientific Writing Academy
Duckburg, Calisota
Donald's Second Affiliation
City, country
donald@swa.edu

A CORRECTNESS ANALYSIS OF NON-INTERACTIVE ZERO-KNOWLEDGE PROOF BASED SECURITY SIMILARITY ESTIMATION SCHEME

In this section, we mainly analyze the correctness of the security similarity estimation scheme based on non-interactive zero-knowledge proofs mentioned in Horse-MinHash. As shown in Figure 1, assume the MinHash signature plaintexts of block f and block g respectively are $Sig(f) = [x_1, x_2, \dots, x_{n-1}, x_n]$ and $Sig(g) = [y_1, y_2, \dots, y_{n-1}, y_n]$. This $ESig$ data structure is utilized to encrypt the elements in MinHash signature plaintexts. For elements x_i and y_i (where $1 \leq i \leq n$), there are five parts in the ciphertexts $ESig(x_i, r_m, p)$ and $ESig(y_i, r_n, p)$. The $isEqual(x_i, y_i, r_m, r_n, p)$ is a method to validate whether the signature elements x_i and y_i are equal on their $ESig$ ciphertexts.

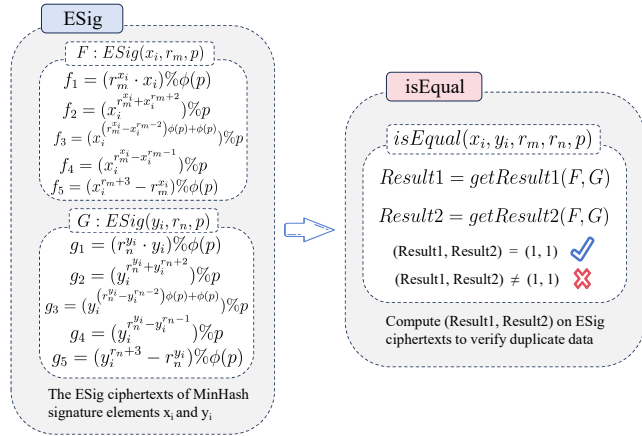


Figure 1: An example of Encrypting Signature Elements Using the ESig Data Structure and Duplicate Element Validation on Ciphertexts Using the isEqual Data Structure

$$Result1 = (((((g_2^{f_1+\phi(p)} \% p \cdot f_3^{-g_1+\phi(p)} \% p) \% p \cdot g_4^{f_5+\phi(p)} \% p \cdot g_3^{f_1+\phi(p)} \% p) \% p \cdot f_2^{-g_1+\phi(p)} \% p) \% p \cdot f_4^{-g_5+\phi(p)} \% p) \% p) \% p \quad (1)$$

$$Result2 = (((((f_2^{g_1+\phi(p)} \% p \cdot g_3^{-f_1+\phi(p)} \% p) \% p \cdot f_4^{g_5+\phi(p)} \% p \cdot f_3^{g_1+\phi(p)} \% p) \% p \cdot g_2^{-f_1+\phi(p)} \% p) \% p \cdot g_4^{-f_5+\phi(p)} \% p) \% p) \% p \quad (2)$$

Assume that client i and client j respectively upload the MinHash signature ciphertext $ESig(f)$ of block f and $ESig(g)$ of block g to the server. The $isEqual$ data structure allows server to estimate the Jaccard similarity between encrypted data uploaded by different users without revealing any additional information. The $getResult1$ and $getResult2$ functions in $isEqual(x_i, y_i, r_m, r_n, p)$ validation methods compute Formula 1 and Formula 2 respectively, and return the results to the variables $Result1$ and $Result2$, where x_i and y_i are elements in the same position in the MinHash signature plaintext of block f and g respectively. If x_i is equal to y_i , then result of Formula 1 is calculated as follows:

$$\begin{aligned} Result1 &= ((x_i^{r_n^{x_i}+x_i^{r_n+2}} \% p)(r_m^{x_i} \cdot x_i) \% \phi(p) \\ &\cdot (x_i^{r_m^{x_i}-x_i^{r_m-2}} \% \phi(p) + \phi(p) - (r_n^{x_i} \cdot x_i) \% \phi(p)) \% p \\ &\cdot ((x_i)^{r_n^{x_i}-x_i^{r_n-1}} \% p)(x_i^{r_m+3}-r_m^{x_i}) \% \phi(p) \% p \\ &\cdot (x_i^{r_n^{x_i}-x_i^{r_n-2}} \% \phi(p) + \phi(p))(r_m^{x_i} * x_i) \% \phi(p) \% p \\ &\cdot ((x_i)^{r_m^{x_i}+x_i^{r_m+2}} \% p) - (r_n^{x_i} * x_i) \% \phi(p) \% p \\ &\cdot (x_i^{r_m^{x_i}-x_i^{r_m-1}} \% p) - (x_i^{(r_n+3)} - r_n^{x_i}) \% \phi(p) \% p \\ &= (x_i^{r_n^{x_i}+x_i^{r_n+2}})(r_m^{x_i} \cdot x_i) + (r_m^{x_i}+x_i^{r_m+2})(-r_n^{x_i} \cdot x_i) \\ &\cdot x_i^{r_m^{x_i}-x_i^{r_m-2}}(-r_n^{x_i} \cdot x_i) + (r_n^{x_i}-x_i^{r_n-2})(r_m^{x_i} \cdot x_i) \% p \\ &= (x_i^0) \% p = 1 \end{aligned}$$

Similarly, we can get that the calculation result of Formula 2 is also equal to 1. This means that the *getResult1* and *getResult2* functions will return (1, 1). If x_i is not equal to y_i , then result of Formula 1 is calculated as follows:

$$\begin{aligned}
 Result1 &= ((y_i^{r_n^{y_i} + y_i^{r_n+2}} \% p)^{(r_m^{x_i} \cdot x_i) \% \phi(p)} \\
 &\cdot (x_i^{r_m^{x_i} - x_i^{r_m-2} \% \phi(p) + \phi(p)} \% p)^{-(r_n^{y_i} \cdot y_i) \% \phi(p)} \% p \\
 &\cdot ((y_i)^{r_n^{y_i} - y_i^{r_n-1}} \% p)^{(x_i^{r_m+3} - r_m^{x_i}) \% \phi(p)} \% p \\
 &\cdot (y_i^{(r_n^{y_i} - y_i^{r_n-2} \% \phi(p) + \phi(p)) \% p})^{(r_m^{x_i} * x_i) \% \phi(p)} \% p \\
 &\cdot ((x_i)^{r_m^{x_i} + x_i^{r_m+2}} \% p)^{-(r_n^{y_i} * y_i) \% \phi(p)} \% p \\
 &\cdot (x_i^{(r_m^{x_i} - x_i^{r_m-1}) \% p})^{-(y_i^{(r_n+3)} - r_n^{y_i}) \% \phi(p)} \% p \\
 &= (x_i^A + y_i^B) \% p
 \end{aligned}$$

$$\begin{aligned}
 \text{Where } A &= (r_n^{y_i} \cdot y_i) \cdot (-2 \cdot r_m^{x_i} + x_i^{r_m-2} + x_i^{r_m+2}) \\
 &\quad - (r_m^{x_i} - x_i^{r_m-1})(y_i^{r_n+3} - r_n^{y_i}) \\
 B &= (r_m^{x_i} \cdot x_i) \cdot (2 \cdot r_n^{y_i} + y_i^{r_n+2} - y_i^{r_n-2}) \\
 &\quad + (r_n^{y_i} - y_i^{r_n-1}) \cdot (x_i^{r_m+3} - r_m^{x_i})
 \end{aligned}$$

Obviously, there is no way to eliminate the random number in *Result1* because x_i is not equal to y_i , *Result1* will not be equal to 1. Similarly, we can obtain the calculation result of Formula 2, *Result2* will not be equal to 1. So the *getResult1* and *getResult2* functions will not return (1, 1).

In cloud storage scenarios, after receiving *ESig(f)* of block f and *ESig(g)* of block g , the server only needs to execute n rounds of duplicate data validations to obtain the Jaccard similarity between the encrypted blocks. If the number of signature elements where $(Result1, Result2) = (1, 1)$ is t , the Jaccard similarity between blocks f and g is estimated as $\frac{t}{n}$, where n represents the number of independent linear transformations $\pi(x) = (a * x + b) \% p$.