

Trabajo Práctico Hashing

Grupo: **G06**

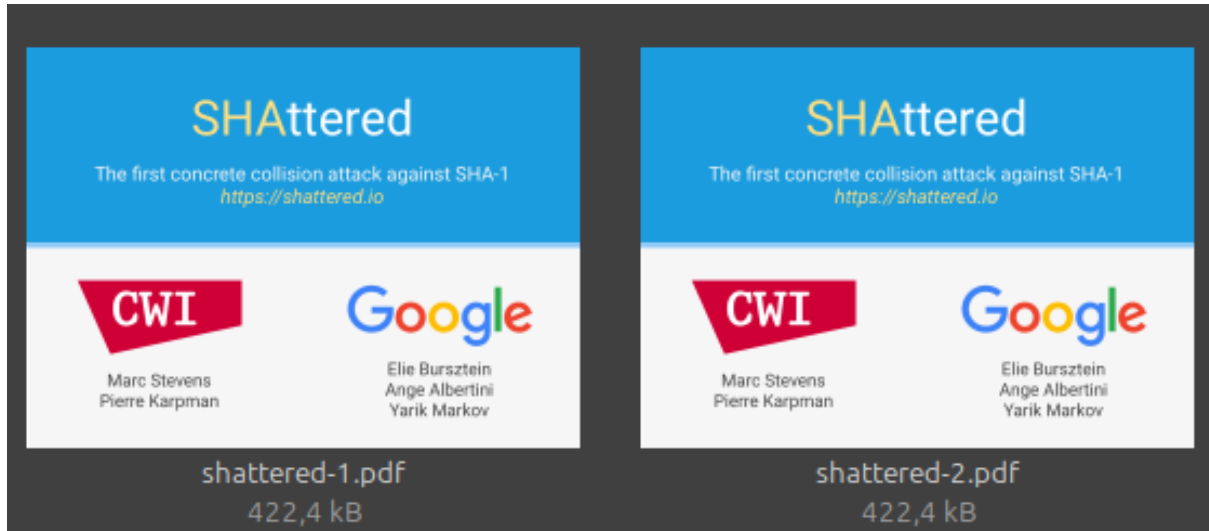
Integrantes:

Nombre	Padrón
<i>Pedro Flynn</i>	<i>105742</i>
<i>Agustina Schmidt</i>	<i>103409</i>
<i>Agustina Fraccaro</i>	<i>103199</i>
<i>Kevin Gadacz</i>	<i>104531</i>
<i>Abraham Osco</i>	<i>102256</i>
<i>Ricardo Luizaga</i>	<i>87528</i>


Ejercicio 1

Usando hash.online-convert.com o cualquier otra herramienta, realizar el hash en SHA1 de los dos archivos que se encuentran en la carpeta SHA1 del campus.

Análisis de los archivos del campus:



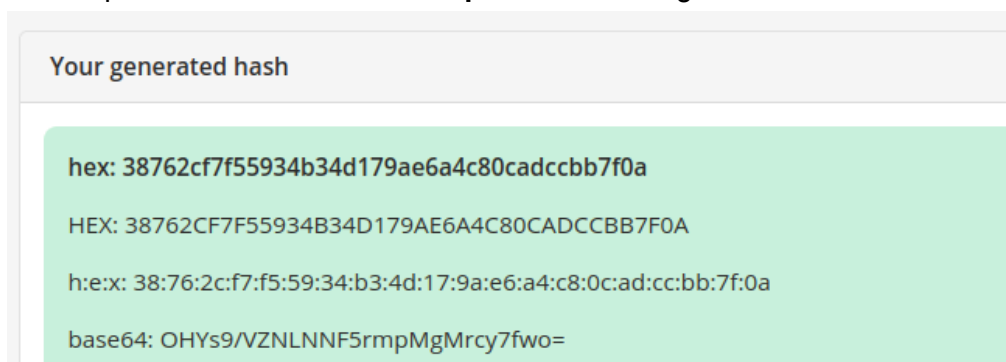
Apriori los archivos en el campus se ven “idénticos” y además pesan exactamente lo mismo (**422,4 kB (422.435 bytes)**)

Nombre:	shattered-2.pdf		Nombre:	shattered-1.pdf
Tipo:	Documento (application/pdf)		Tipo:	Documento (application/pdf)
Tamaño:	422,4 kB (422.435 bytes)		Tamaño:	422,4 kB (422.435 bytes)

Realizamos el hash usando la siguiente página:

<https://hash.online-convert.com/sha1-generator>

Para el primer archivo **shattered-1.pdf** usando el algoritmo SHA1 se obtuvo:



hex: 38762cf7f55934b34d179ae6a4c80cadccbb7f0a

Para el segundo archivo **shattered-2.pdf** usando el algoritmo SHA1 se obtuvo:

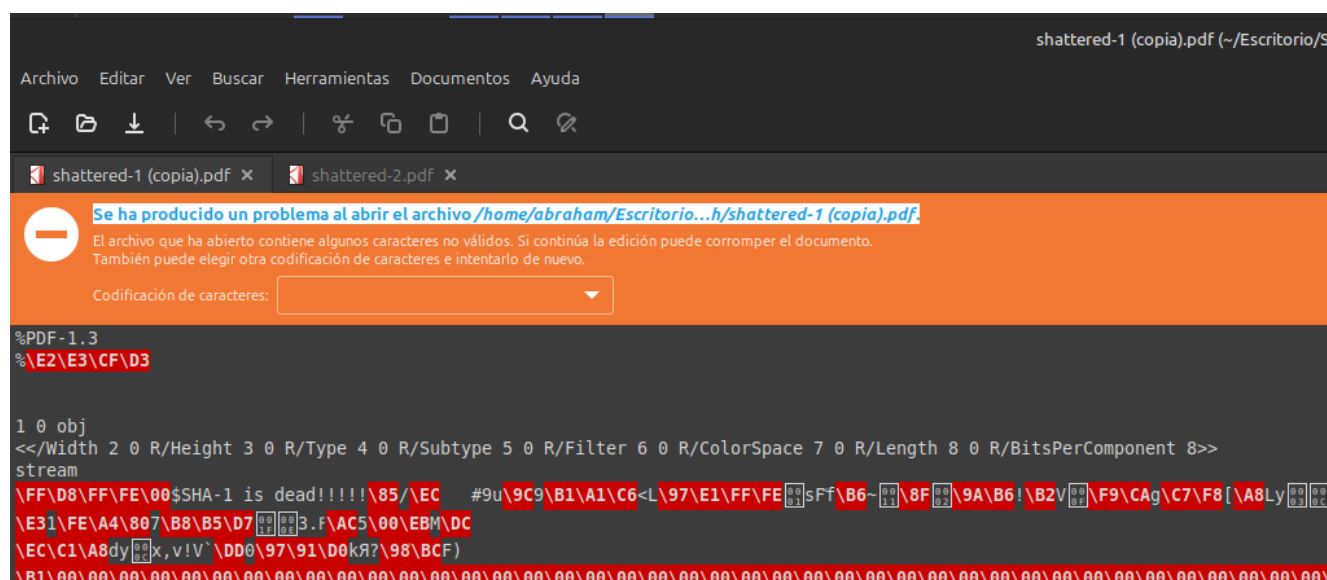
Your generated hash

```
hex: 38762cf7f55934b34d179ae6a4c80cadccbb7f0a
HEX: 38762CF7F55934B34D179AE6A4C80CADCCBB7F0A
h:e:x: 38:76:2c:f7:f5:59:34:b3:4d:17:9a:e6:a4:c8:0c:ad:cc:bb:7f:0a
base64: OHYs9/VZNLNNF5rmpMgMrcy7fwo=
```

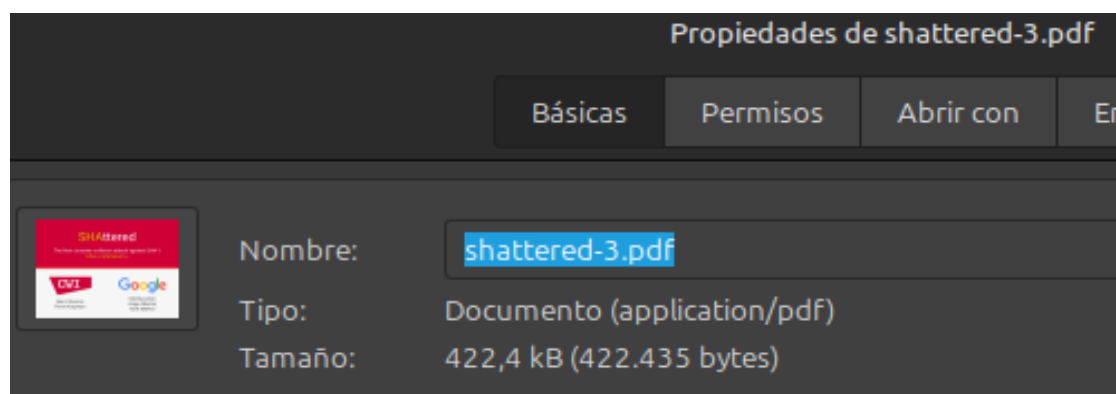
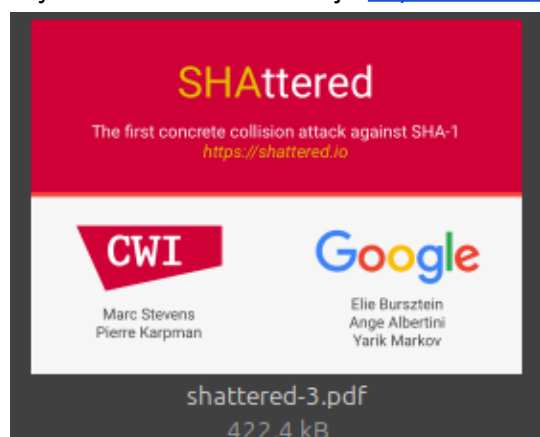
hex: 38762cf7f55934b34d179ae6a4c80cadccbb7f0a

Obtuvimos que ambos hashes son exactamente iguales.

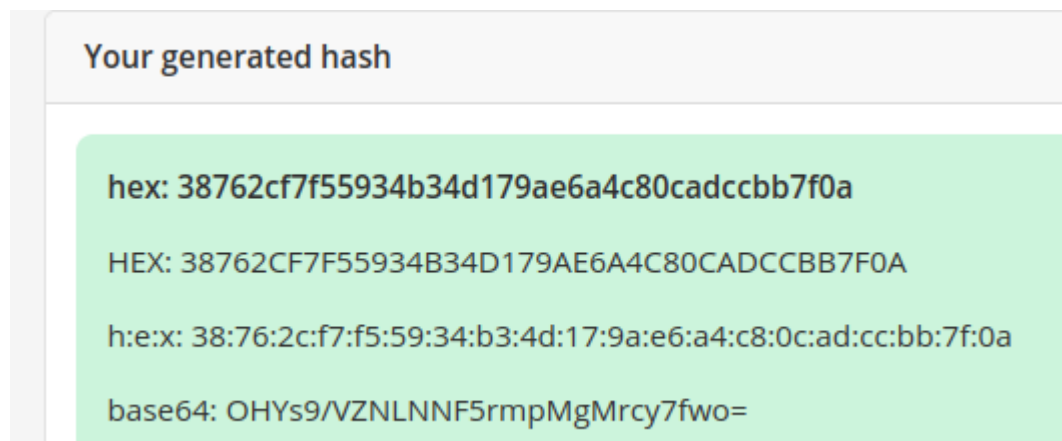
Curiosidad: Si visualizamos el archivo **shattered-1.pdf** usando el editor de texto de linux (Que nos mapea el código en hexa del archivo a string) podemos ver que Existe un **\$SHA-1 is dead!!!!**:



Ahora si entramos al link que describe el los archivos: <https://shattered.io/> Y descargamos el segundo pdf cuyo fondo es de color rojo <https://shattered.io/static/shattered-2.pdf> lo nombramos como shattered-3.pdf.



Y obtenemos su hash usando SHA1:

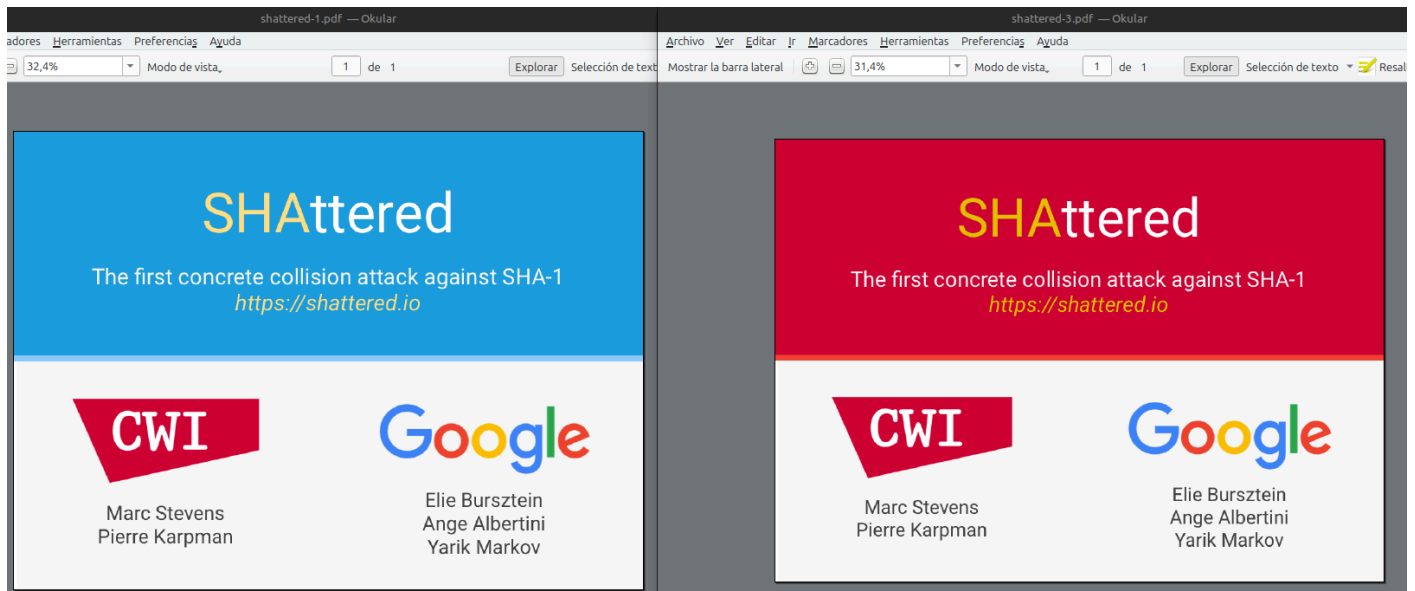


Encontramos que tenemos el mismo hash que los archivos shattered-1.pdf shattered-2.pdf :
hex: 38762cf7f55934b34d179ae6a4c80cadccbb7f0a

1) ¿Qué propiedad de las funciones de hash no se cumple?

Del análisis anterior es evidente que se **violan** 3 propiedades de las funciones de hash:

1. **Resistencia a la preimagen:** Hemos encontrado en la pagina <https://shattered.io/> un archivo que tiene el mismo hash que los archivos del campus **shattered-1.pdf shattered-2.pdf**.
2. **Resistencia a la 2º preimagen:** Idem al anterior hemos encontrado un archivo **shattered-3.pdf** **distinto** a **shattered-1.pdf** y **shattered-2.pdf** y obtuvimos exactamente el mismo hash.



3. **Resistencia a las colisiones:** Hemos encontrado 2 archivos (entradas) distintos con la misma salida (el mismo hash).

2) ¿Qué problemas se podrían generar?

Compromiso de la integridad de los datos: Si se pueden crear dos archivos diferentes que generen el mismo hash, un atacante podría modificar un archivo (como un contrato o documento importante) manteniendo el mismo hash, lo que puede engañar a los sistemas de verificación de integridad.

Vulnerabilidad en sistemas de firma digital: Las firmas digitales dependen de la función de hash para garantizar que los datos no han sido alterados. Si dos archivos pueden tener el mismo hash, un atacante podría sustituir un documento firmado legítimamente con otro sin que se note el cambio.

3) Medidas para mitigarlo

Actualizar las funciones de hashing usando algoritmos más seguros: Dado que SHA-1 ha sido roto (como lo demuestran los ataques de colisión), es esencial migrar a algoritmos de hash más seguros, como SHA-256 o SHA-3, que ofrecen mayor resistencia a ataques de colisión.

Uso de funciones hash resistentes a colisiones: En sistemas críticos (como firmas digitales o almacenamiento de contraseñas), es recomendable usar funciones hash diseñadas específicamente para resistir colisiones.

Ejercicio 2

¿Cómo se usa un HASH en una pericia?

En una pericia informática, los HASH se utilizan para garantizar la integridad de las evidencias recolectadas. Se extraen HASH de los archivos y otros datos relevantes para asegurar que no han sido modificados desde el momento en que fueron recolectados. Esto es especialmente importante en la cadena de custodia de las evidencias, ya que permite verificar que la información se mantuvo intacta durante todo el proceso.

El uso de HASH asegura que los datos se pueden reproducir y autenticar, lo que es crucial si las evidencias se presentan en un tribunal. Según el documento, es necesario calcular los hashes de las evidencias como parte de la documentación de la cadena de custodia para asegurar que no se haya modificado ninguna imagen o archivo

¿Qué HASH usaría para una pericia informática?

En el análisis forense, el documento menciona que, aunque el MD5 es ampliamente utilizado, presenta el problema de colisiones (es decir, diferentes archivos podrían generar el mismo hash) (Esto mismo puede suceder con SHA-1), lo que puede cuestionar la validez de las pruebas. Por lo tanto, se recomienda utilizar alternativas más seguras como SHA-256 o SHA-512, que son más robustas y evitan este tipo de colisiones