Alex Mills
9/18/2015
CS 465

OBJECTIVE

To understand the difference between collision and pre-image attacks. To demonstrate the ability to design and conduct an experiment and then present the results in a technical manner. To test the theoretical cost of pre-image and collision attacks through experimentation.

PROCEDURE

1. Create two functions that will respectively run the collision and pre-image attacks
    a. Collision attack
        i. Have a variable that contains the number of attempts to find a matching hash.
        ii. Have a map that contains all of the different hashes that have been generated. This map will be used to match each of the newly generated hashes to the already generated hashes.
        iii. Run a while loop until two matching hashes are found.
        iv. Generate a random string that will be hashed on every iteration of the while loop.
        v. Check if the hash has already been generated, if not then store it, if it has then break the while loop and log the results.
        vi. Create a for loop around the while loop that will run a set number of times in order to find an average number of attempts.
    b. Pre-Image attack
        i. First create a random string and get the hash that will be used as the pre-image.
        ii. Have a variable that counts the number of attempts to find a matching string.
        iii. Run a while loop until a matching hash is found.
        iv. Inside the while loop generate a random string and hash it.
        v. Use the new hash to match against the pre-image hash.
        vi. Ensure the generated strings are not the same.
        vii. Ensure that the generated strings are not null.
        viii. When a matching hash has been found, log the results.
2. Find and use a hash algorithm that produces a 40 character hex output.
3. Create a function that will change the hex characters of the hash output to binary.
4. Truncate the number of bits returned from the hash function to match the number of bits determined at the outset of the experiment.
5. Run the collision attacks for 1, 5, 10, 15, 20, 25, 30, and 35 bits. Document the results.
6. Run the pre-image attacks for 1, 5, 10, 15, 20, 25, 30, and 35 bits. Document the results.

RESULTS

COLLISION

Number of Bits	Average of 100	Theoretical	% Difference
1	3.54	1.41	86.06
5	8.65	5.657	41.84
10	40.74	32	24.03
15	242.68	181.019	29.11
20	1254.03	1024	20.2
25	7339.03	5792.619	23.55
30	42971.81	32768	26.94
35	217536.48	185363.80	15.97

The collision attack followed the theoretical number of attempts being 2^(n/2). Overall the percent difference for meaningful numbers of bits is not over 30%. The collision attack was run 100 times, while this is only one example of the collision attack, the percent difference would range above and below the theoretical average. Overall the measured averages were higher than the theoretical average. This can be due to the somewhat fixed size of the random string generator and the fixed 26 lowercase letters.

PRE-IMAGE

Number of Bits	Average of 100	Theoretical	% Difference
1	1.76	2	12.77
5	29.8	32	7.12
10	971.81	1024	5.23
15	28287.82	32768	14.68
20	1399532.75	1048576	28.67
25	-	-	-
30	-	-	-
35	-	-	-

The pre-image attack was also run 100 times for each of the number of bits. The theoretical average is 2^n, being different by orders of magnitude than the collision attack. When running the pre-image attack with 20 bits, processing time was much too long to wait for more than 5 completed rounds. The average number for the 20 bit attack is the average of the 5 rounds. A pre-image attack of 25 bits was attempted but after 45 minutes with no results the attack was discontinued. The average number of attempts up to the 20 bit attack were less than the theoretical average number of attempts. The 20 bit attempt was the first attack where the average was greater than the theoretical average. Should the 20 bit attack have gone for the full 100 rounds it is possible that the average number of attacks would be less than the theoretical number.

It was discovered during development and testing that the pre-image attack was showing an average of 2,500 - 3,500 per each attack. After investigation, it was shown that the random string generator was producing the same string. The random string generator would produce the same string between every 100 and 5,000 strings. This was producing a false positive in that the new string and the pre-image were hashing to the same value simply because they were the same. This may also be a reason that the average number of collision attacks for the different number of bits was higher than the theoretical number, because the same random strings would be used multiple times in the search for a collision.

CONCLUSION

Through experimentation it is proved that the theoretical averages for pre-image and collision attacks are the true averages. For the pre-image attacks that could be run in a reasonable amount of time, the averages were slightly under the theoretical average being anywhere between 5 and 28 percent different. For the collision attacks the averages were anywhere between 15 and 29 percent. The first two collision attacks with 1 bit and 5 bits are such an easy number of bits to match that theoretically they could both be accomplished in less than 6 attempts. The measured average number of attacks were higher based on how the random string generator created strings to be hashed the limited number of characters used in the generated string.

RESOURCES

SHA1 hash algorithm
http://www.movable-type.co.uk/scripts/sha1.html

Base64 to binary function
http://vabate.com/convert-a-base64-encoded-string-to-binary-with-javascript/

CryptoJS - SHA1 hash function
https://code.google.com/p/crypto-js/#SHA-1

CryptoJS for dummies
http://www.davidebarranca.com/2012/10/crypto-js-tutorial-cryptography-for-dummies/

Percent Change (new- old/ old)
http://www.mathsisfun.com/data/percentage-difference-vs-error.html

APPENDIX

All code can be found on my github account at http://github.com/paneMrazek.