Spring 2019 Cryptography and Network Security
Final Project Proposal
Next Generation Hashing: Robustness of Neural Network Generated Hash Function
DUE DATE: 5/7/2019, 23:59

B03505052 李吉昌 R07944058 陳鵬宇 B05902001 廖彥綸 B06902119 張原豪 B06902047 陳彥

Problem Description

People nowadays often use cryptographic hash functions for message authentication or ciphertext signing. However, it is declared that some hash functions designed for this purpose are actually insecure (e.g., MD5, SHA-*). These hash functions poses problem for length extension attack, dictionary attack, and can not withstand collision attack. Therefore, we would like to survey if there is another way to construct hash functions, and if so, how resilient it is against common attacks. Moreover, we would like to find out the pros and cons of such construction and ways to improve it.

Several studies attempted to use neural network to construct cryptographic hash functions. Our goal is to check whether it is feasible to construct hash functions this way. We also need to check whether the hash function satisfies one-wayness, weak collision resistance and strong collision resistance. In addition, we attempt to come up with several attacks to test its resilience and discuss whether publishing or concealing the NN model is best for such practice.

Related Work

Several researches for cryptographic hash function construction based on neural network has been conducted. These proposals make use of various techniques like DNN[1] or RNN[2] to generate a specific hash function by the fixed length of encoded input message. Previously, diffusion based on chaotic map had also been proposed[3][4]. Extensional research such as Tree Parity Machine (TPM), which uses NN to design special exchanged key[5], were mentioned as well.

However, some of the proposals mentioned above lack practical implementation on proving the security against known data and reverse attack. Also, the consideration on revealing model is wanting. Due to these reasons, the scheduled task in timeline is arranged in hope of resolving such problems. Most of the papers focus merely on demonstrating the robustness of NN-generated hash function, hence ignorance on the comparison with already existed hash functions. We state that more experiments must be conducted to ensure the strengths and weaknesses of NN-generated hash function, including its security and possible attacks toward it. The plan is provided below for further explanation of the experiment.

Plan

1. Generating a model as hash function using neural network based on papers in the reference. Improvement of security or performance from existed model is expected. On the other hand, we try to select several attacks to break or cause damage to the original model. The defense methods are also in the wishlist once an attack succeeded.

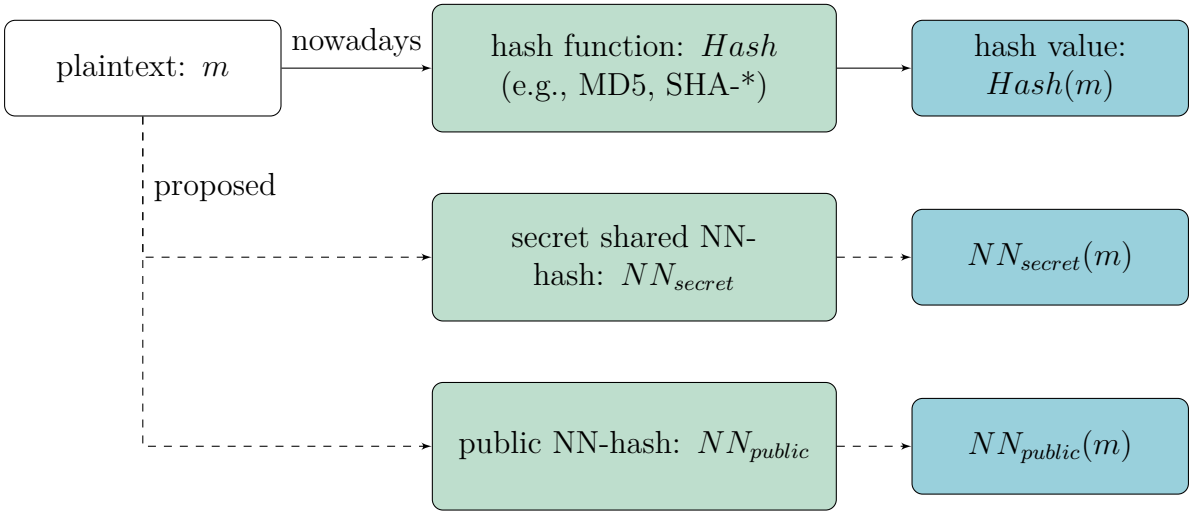
2. Serveral experiments are used for evaluating the result of our model and original one:

   (1) Analytical experiment on hash function diffusion between two models

(2) Using the computed loss and compare it with the randomly generated loss

(3) Analyzing the attacks on the lecture note and seeing if it is effective on our model

(4) Feasibility of reverse attack in case of published model



Timeline

Time	Work
5/10	Establishing NN, RNN hash function, performance testing.
5/17	Evaluating the feasibility of the model via constructing diverse collision attack.
5/24	Eexploring diffusive method and using different compression method (padding, known-compression).
5/31	Demonstrating the attack and defense data-collecting reverse attack and man in the middle attack...
6/7	Summarizing to-date job, proposing future related work and limitation.

Deliverables

1. Demoing model.
2. Showing the attack we have tried and its results.
3. Showing the diffusion experiment and its results.

References

[1] Michal Turčaník, Hash function generation based on neural networks and chaotic maps, Oct 2016, KIT
[2] Michal Turčaník, Using recurrent neural network for hash function generation, 2017, AE
[3] V. R. Kulkarni, Shaheen Mujawar and Sulabha Apte, HASH FUNCTION IMPLEMENTATION USING ARTIFICIAL NEURAL NETWORK, 2010, IJSC
[4] Liew Pol Yee' and Liyanage C. De Silva', Application of MultiLayer Perceptron Network as a One-way Hash Function, 2002, IEEE
[5] Vineeta Soni, Mrs. Sarvesh Tanwar, Prof. Prema K.V., Implementation of Hash Function Based On Neural Cryptography, 2014, IJCSMC