SHA-RNN

SHA-CNN: Chaotic Neural Network

[Abstract]

1 Introduction

2 Related Works

2.1 Sponge Function

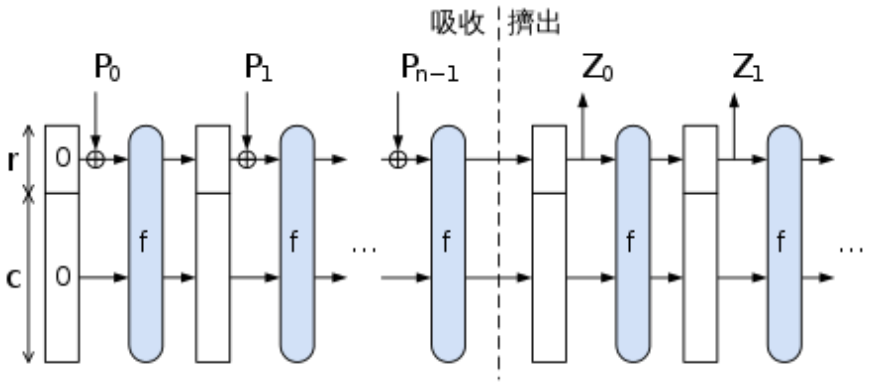

Figure 2.1.1: 海绵函数的示意图
Credit: Wikipedia

海绵函数(Sponge Function) [1] 是一种算法，它可以接受任意长度的输入比特流，得到任意长度的输出。它的参数为单次输出比特长度 r，隐藏状态长度 $r+c$，状态转移函数 f。其工作流程可以分为 **吸收(Absorb)** 与 **挤出(Squeeze)** 两个阶段，其工作流程如图 2.1.1 所示，描述如下。

在吸收阶段，海绵函数单次接受长度为 r 的消息分组，与上个阶段的隐藏状态 r 比特进行异或，将异或后的隐藏状态经过状态转移函数 f 得到新的隐藏状态。在挤出阶段，我们每次从隐藏状态中提取 r 比特，然后将隐藏状态再经过状态转移函数 f，如此往复，直到我们得到足够长的消息摘要为止。

2.2 CNN: Chaotic Neural Network

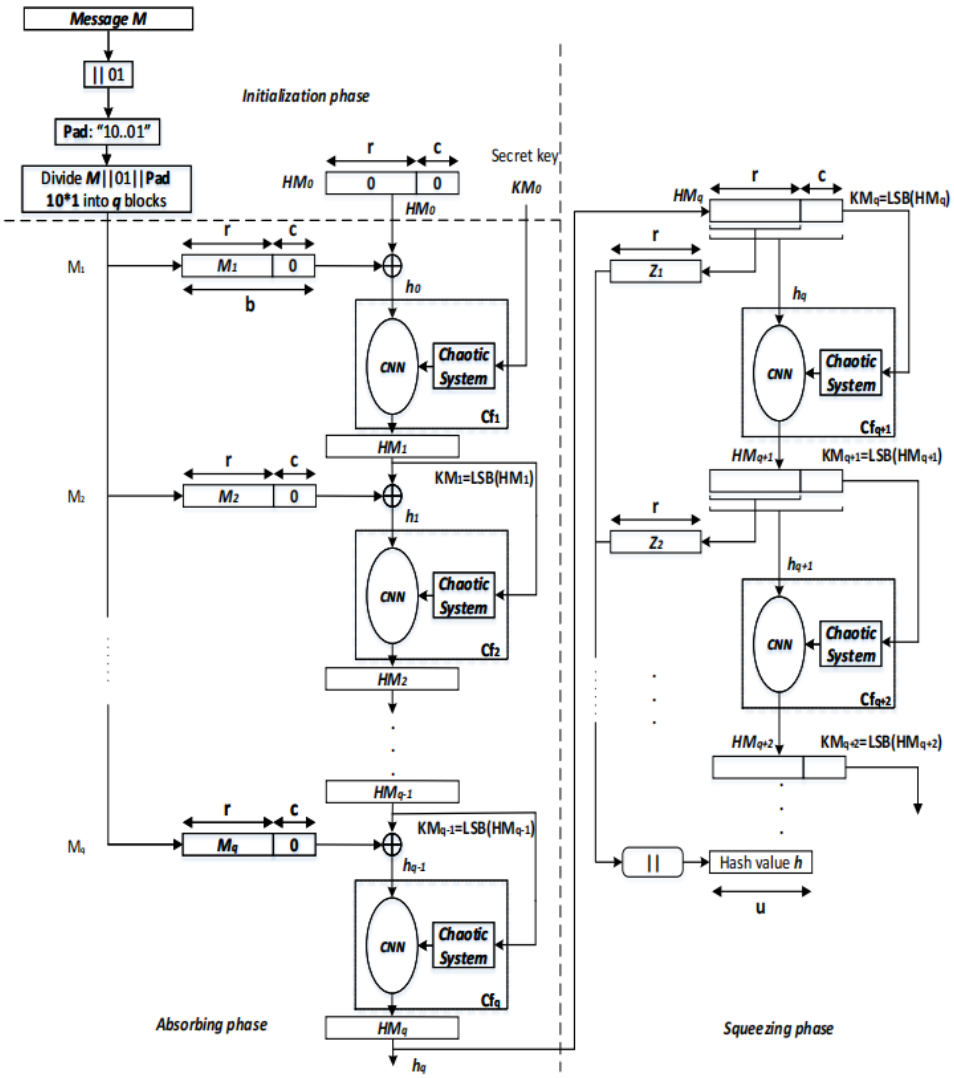

Figure 2.2.1 Keyed-Sponge CNN hash functions

Credit: Citation [2], Figure 3.

引用文献 [2] 提出了两种基于密钥的哈希函数，其采用了海绵函数架构，使用了 **混沌神经网络(Chaotic Neural Network)** 作为海绵函数的状态转移函数 f。其基本运作模式如图 2.2.1 所示，接下来我们对其运作阶段及状态转移函数 f 进行简单介绍。

首先我们来对其运作的三个阶段进行简单介绍。**(1)** 在初始阶段，我们假设输入密钥为 K，将隐藏状态 $HM_0 \leftarrow 0^{r+c}$，密钥 $KM_0 \leftarrow K$，并对输入做适当填充后分组。**(2)** 在第 q 个吸收阶段，将本阶段输入消息分组 M_q 与隐藏状态 HM_{q-1} 的前 r 个比特做异或之后，我们将隐藏状态 h_{q-1} 与混沌发生参数 KM_{q-1} 输入混沌神经网络系统之中（包括一个混沌神经网络和一个混沌系统，我们会在后续进行介绍）。**(3)** 在第 q 个挤出阶段，我们输出隐藏状态 h_{q-1} 的前 r 比特输出，作为当前阶段的消息摘要输出，然后将隐藏状态 h_{q-1} 与上一阶段的生成密钥 KM_{q-1} 输入混沌神经网络中，得到下一阶段的隐藏状态 HM_q。

这里的 **混沌神经系统** 作为海绵函数 f 的状态转移函数，承担着接受密钥，将隐藏状态混淆的同时使得输出结果尽可能均匀的作用。**混沌系统(Chaotic System)** 接受上个阶段的输出密钥 KM_{q-1} 作为发生参数，产生均匀的输出作为后续 **混沌神经网络(Chaotic Neural Network, CNN)** 各层的权重（Weights）与偏差（Biases）。这里的混沌神经网络采用了简单的全连接层的机制。

2.3 DSTMap: Discrete Skew Tent Map

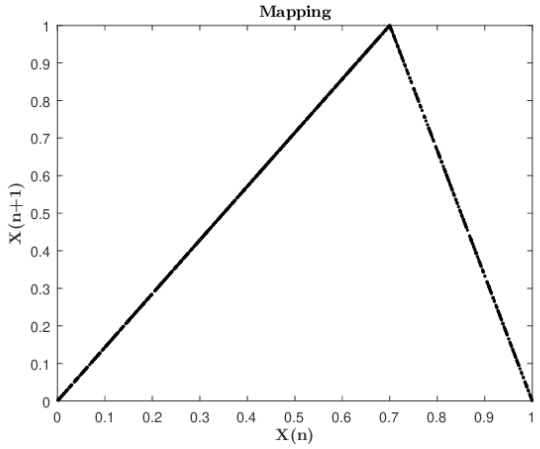

Figure 2.3.1 Skew Tent Mapping
Credit: Research Gate

我们这里首先引入 Skew Tent Map [3] 的概念，它被广泛地应用于混沌系统的生成当中。Skew Tent Map 是一种迭代函数，它接受上一状态输入 $X(n)$，将其映射为：

$$X(n+1) = \begin{cases} \frac{\mu}{Q}x, \ X(n) \leq Q, \\ \frac{-\mu}{1-Q}(x-Q)+\mu, \ X(n) > Q. \end{cases}$$

对于大部分输入，通过足够多的迭代步，可以保证这个函数生成的序列是混沌的。而我们这里介绍其离散化的版本，即对于输入 $1 \leq X(n-1) \leq 2^n - 1$：

$$\text{DSTmap}(X(n-1), Q) = \begin{cases} 2^N \times \frac{X(n-1)}{Q} & \text{if } 0 < X(n-1) < Q \\ 2^N - 1 & \text{if } X(n-1) = Q \\ 2^N \times \frac{2^N - X(n-1)}{2^N - Q} & \text{if } Q < X(n-1) < 2^N \end{cases}$$

我们将在混沌系统中引入这个函数以保证生成序列的随机性。

3 SHA-RNN

4 Evalutaion

5 References

[1] Bertoni G, Daemen J, Peeters M, et al. Sponge functions[C]//ECRYPT hash workshop. 2007, 2007(9).

[2] Abdoun N, El Assad S, Manh Hoang T, et al. Designing two secure keyed hash functions based on sponge construction and the chaotic neural network[J]. Entropy, 2020, 22(9): 1012.

[3] Hasler M, Maistrenko Y L. An introduction to the synchronization of chaotic systems: coupled skew tent maps[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1997, 44(10): 856-866.