SHA-RNN

Recurrent Neural Network with Chaotic System for Hash Functions

Anonymous Authors

[**摘要**] 在这次作业中我们提出了一种新的 Hash Function —— SHA-RNN。其以海绵结构为基础，融合了混沌系统与循环神经网络的特点，可以接受任意长度的串作为输入，并生成长度为 80 比特的消息摘要。我们会对我们的架构进行详细表述，并给出在时间及安全性上的自评估结果。最后，我们会提供完整的实验框架与使用说明。

GitHub Repo 地址：https://github.com/Ashitemaru/Sharnn

1 Introduction

哈希函数（Hash Functions）是一种将任意长度的消息生成固定长度摘要的映射，其在消息认证、数字签名、身份验证等方面具有十分重要的应用。一般地，我们要求哈希函数具有有效性与安全性，前者是说哈希函数的运作效率要足够快，而后者则对哈希函数的抗原象攻击、抗第二原象攻击等等方面具有一定要求。

混沌系统（Chaotic System）的概念最先在物理学中的运动学领域被提出，它是一种非线性系统，其中的值存在着貌似随机的不规则运动，其行为表现为不确定性、不可预测。有部分可以迭代计算的函数可以作为混沌系统的生成函数，比如帐篷函数（Tent Map）。由于混沌系统具有的给定初值输入后，其后生成序列值不可预测，或者说近似服从均匀分布的特点，其可以被用来作为 Hash 函数的重要组成部分。[1]

循环神经网络（Recurrent Neural Network）是一种以序列数据为输入，在序列时序信息的维度上迭代，所有循环单元按链式进行连接的一种递归神经网络。循环神经网络的优势之一，在于其能够在某种程度上“记忆”来自上一个时间状态的信息。如果不同消息分组输入循环神经网络的顺序有先后之别，他们对于同一个“神经元”的作用也有不同的区别。而这种区别，往往是难以用计算追溯到的，也是其相对于简单的全连接层的优势所在。

综合以上三个概念，我们提出了 **SHA-RNN**，一种新的基于密钥的哈希函数。其采用海绵函数作为基本架构，在每次迭代中使用混沌系统产生近似均匀分布的值作为循环神经网络的权重和偏差。

本篇报告将分为以下几部分。在第 2 节，我们会对我们设计哈希函数时所运用到的关键概念进行简要的回放与讲解。在第 3 节，我们会给出我们所设计的哈希函数的具体参数与详细运作流程。在第 4 节，我们会对我们所设计的哈希函数进行分析，给出在时间效率上和安全性检测上的自评估结果。在附录中，我们会对我们编写的实验框架进行详细的介绍，包括代码介绍与使用方法介绍。

2 Related Works

2.1 Sponge Function

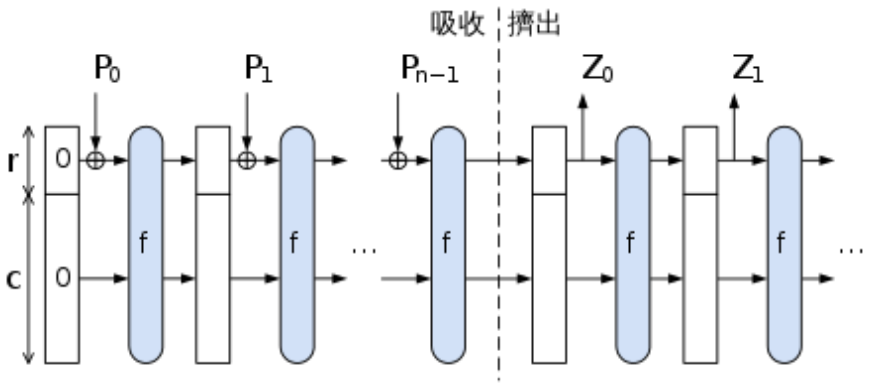

Figure 2.1.1: 海绵函数的示意图
Credit: Wikipedia

海绵函数(Sponge Function) [2] 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 Chaotic Neural Network

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。

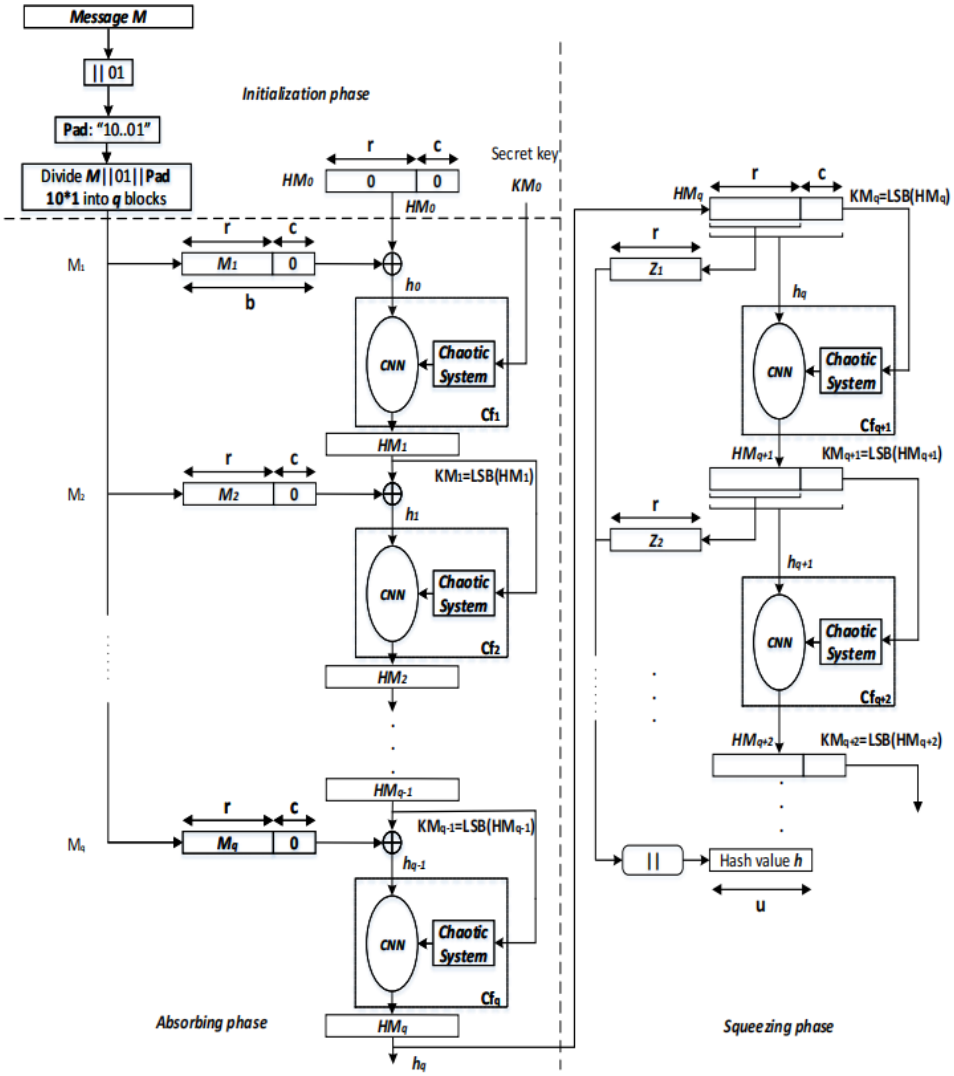

Figure 2.2.1 Keyed-Sponge CNN hash functions

Credit: Citation [3], Figure 3.

2.3 Discrete Skew Tent Map

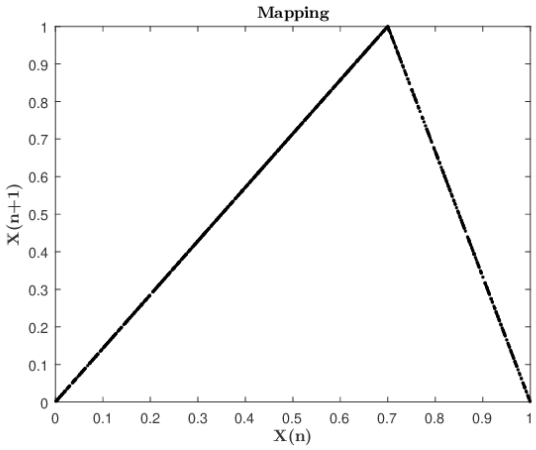

Figure 2.3.1 Skew Tent Mapping

Credit: Research Gate

我们这里首先引入 Skew Tent Map [4] 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：

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

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 $1 \le X(n-1) \le 2^n - 1$：

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

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

我们

4 Evalutaion

5 References

[1] LIU J, FU X. Spatiotemporal chaotic one-way hash function construction based on coupled tent maps[J]. Journal on communications, 2007, 28(6): 34.

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

[3] 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.

[4] 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.

6 附录：实验框架介绍