В результате использования, разработанного с помощью библиотеки Moviepy модуля конвертации, полная обработка (декодирование, парсинг и транскодирование) двух видеофайлов заняла 53 секунды. Однако из-за увеличенной сложности реализации полученные итоговые файлы уступают в качестве файлам, созданным с помощью библиотеки MoviePY.

Список использованных источников

1. Обухова, Е. В., Н. П. Шутько Сравнительный анализ алгоритмов видеокодирования THEORA, MPEG-4 и H.263 / Е. В. Обухова, Н. П. Шутько // Сборник материалов международной молодежной научно-практической конференции студентов, аспирантов и молодых ученых, Махачкала, 19–20 ноября 2024 г. – Махачкала, Типография ФОРМАТ, 2024. – С. 71-73.

2. Журавлев, А. А. Сравнительный анализ эффективности программных инструментов для разбивки видео на кадры на примере области оценки качества дорожной поверхности / А. А. Журавлев, К. А. Аксенов // ИВД. – 2024. – №3 (111).

УДК 004.822

Е.В. Обухова
Белорусский государственный технологический университет
Минск, Беларусь

ВЛИЯНИЕ СТРУКТУРНЫХ ПАРАМЕТРОВ НА ЭФФЕКТИВНОСТЬ СИНХРОНИЗАЦИИ ДРЕВОВИДНЫХ МАШИН ЧЕТНОСТИ В УСЛОВИЯХ СЕТЕВЫХ ЗАДЕРЖЕК

Аннотация. *В работе исследуется влияние структурных параметров древовидных машин четности (ДМЧ) на эффективность процесса синхронизации. На основе полученных экспериментальных данных построены зависимости количества итераций и скорости генерации ключа от конфигурации нейросети. Сформулированы рекомендации по выбору оптимальной конфигурации ДМЧ .*

E.V. Abukhova
Belarusian State Technological University
Minsk, Belarus

THE IMPACT OF STRUCTURAL PARAMETERS ON TREE PARITY MACHINE SYNCHRONIZATION EFFICIENCY UNDER NETWORK DELAYS

Abstract. *This paper investigates the impact of the structural parameters of Tree Parity Machines (TPM) on the efficiency of the synchronization process. Based on the obtained experimental data, the dependencies of the iteration count and the key generation rate on the neural network configuration are analyzed. Recommendations for selecting the optimal TPM configuration are formulated.*

Древовидные машины четности [1] (далее – ДМЧ) относятся к специализированному семейству искусственных нейронных сетей. применяемых в нейрокриптографии. В таких моделях обновление синаптических весов осуществляется по правилам взаимного обучения (например, по правилу Хебба), что позволяет двум удаленным сторонам синхронизировать свои внутренние состояния и выработать общий секретный ключ без его прямой передачи по открытому каналу.

Структурно ДМЧ включает входной слой нейронов, скрытый ассоциативный слой и одиночный выходной нейрон. Архитектура определяется тремя структурными параметрами:

1. K – количество скрытых нейронов в промежуточном слое.

2. N – размерность входных сигналов, поступающих на каждый скрытый нейрон. Входные значения x_{kn} формируют двоичный вектор, где каждый элемент принимает значения -1 или 1, обеспечивая высокую дискретизацию входного сигнала.

3. L – параметр, ограничивающий диапазон значений весовых коэффициентов $w_{ij} \in \{-L, \dots, L\}$. Данный параметр определяет объем пространства ключей и сложность подбора весов методом грубой силы.

Выход нейросети формируется как произведение сигнальных значений скрытого слоя, а сам процесс взаимного обучения построен таким образом, что обе стороны, обмениваясь результатами вычислений, корректируют свои веса до полного совпадения.

Описанные ранее параметры K, N и L в значительной степени влияют не только на криптостойкость, но и на производительность системы [2]. Для экспериментальной оценки этого влияния был разработан ряд функций и классов, позволяющих моделировать взаимодействие двух удаленных сторон и исследовать динамику синхронизации их весовых коэффициентов.

Далее будут разобраны основные части класса TP, реализующего архитектуру единичной древовидной машины четности и отвечающего за определение структурных параметров,

которые ранее были описаны теоретически. Так, инициализация весовых коэффициентов w_{ij}, ограниченных параметром L для задания границы пространства поиска, в разработанном классе производится случайным образом на основе равномерного распределения целых чисел. В программной реализации матрица весов W размерности $K \times N$ генерируется с помощью метода np.random.randint() (листинг 1).

```
    def __init__(self, K=K, N=N, L=L, seed=None):
            self.K = K
            self.N = N
            self.L = L
            if seed is not None:
                np.random.seed(seed)
            self.weights = np.random.randint(-L, L+1,
size=(K, N)).astype(float)
```

Листинг 1 – Инициализация весовой матрицы ДМЧ

Процесс синхронизации в созданном классе основан на правиле Хебба. Так, обновление весов происходит только при совпадении выходных значений обеих машин ($\tau_A = \tau_B$). Внутри алгоритма реализована функция ограничения, которая удерживает веса в пределах параметра L, что важно для обеспечения конечности пространства ключей и сходимости алгоритма (листинг 2).

```
    def update(self, x, tau_self, tau_other):
      if tau_self == tau_other:
       for k in range(self.K):
        if self.sigma[k] == tau_other:
         for j in range(self.N):
          self.weights[k][j] += self.sigma[k] * x[k][j]
          self.weights[k][j] = np.clip(self.weights[k][j],
-self.L, self.L)
```

Листинг 2 – Обновление весов

Далее, с помощью созданных программных файлов была проведена серия тестов, в ходе которых значения структурных параметров последовательно изменялись, после чего фиксировались такие параметры, как количество итераций до синхронизации машин и время синхронизации. Впоследствии на основе полученных данных высчитывалось среднее количество итераций до полной синхронизации машин для каждого изменённого значения, а также количество бит секретного ключа, вырабатываемых в секунду реального времени (Bit Rate). Результаты тестов представлены в таблицах 1, 2, 3.

Таблица 1- Влияние числа нейронов (*K*) при *N*=4, *L*=2

K	N	L	Среднее число итераций	Bit Rate (бит/сек)
2	4	2	45,23	14,1427
3	4	2	108,33	9,1009
4	4	2	147,62	8,2365
5	4	2	187,15	11,7938

Таблица 2- Влияние размерности входа (*N*) при *K*=5, *L*=2

K	N	L	Среднее число итераций	Bit Rate (бит/сек)
5	2	2	145,43	7,7959
5	3	2	203,45	7,2092
5	4	2	187,15	11,7938
5	5	2	179,1	24,9901

Таблица 3- Влияние диапазона весов (*L*) при *K*=5, *N*=5

K	N	L	Среднее число итераций	Bit Rate (бит/сек)
5	5	2	179,1	24,9901
5	5	3	465,62	7,8043
5	5	4	1111,92	5,6113
5	5	5	3446,2	1,8268

Таким образом, исходя из полученных в ходе экспериментов данных, представленных в таблице 1, можно выявить, что увеличение числа скрытых нейронов (*K*) приводит к росту числа итераций необходимых для синхронизации машин, так как согласовать выходы большего числа нейронов становится сложнее. Несмотря на то, что длина ключа растет пропорционально *K*, время, затрачиваемое на синхронизацию, растет быстрее, что снижает скорость генерации ключа. Зависимость числа итераций и скорости генерации ключа от параметра *K* представлена на рисунке 1.

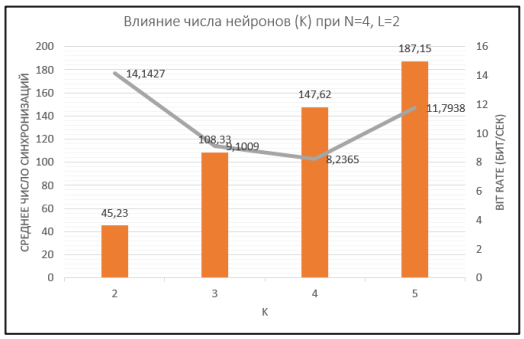

Рис. 1 – Зависимость числа итераций и скорости генерации ключа от параметра *K*

Как видно из таблицы 2, повышение значения *N* практически не обуславливает рост числа итераций, но линейно увеличивает длину вырабатываемого ключа, в результате чего эффективность системы и способность ДМЧ к обучению возрастает: при переходе от *N*=2 к *N*=5 скорость генерации ключа выросла более чем в 3 раза. Зависимость числа итераций и скорости генерации ключа от параметра *N* представлена на рисунке 2.

Рис. 2 – Зависимость числа итераций и скорости генерации ключа от параметра *N*

Диапазон значений весов (*L*) оказывает наибольшее влияние на сложность синхронизации. С увеличением параметра *L* расширяется интервал возможных весовых коэффициентов, что приводит к росту сложности перебора для атакующего в случае проведения атаки на систему, однако в это же время растет и число итераций для обычных пользователей. Данные, представленные в таблице 3, показывают, что при увеличении параметра *L* с 2 до 5 скорость генерации ключа упала с 24.99 до 1.82 бит/с. Это свидетельствует о том, что большие значения *L* следует использовать только при критических требованиях к безопасности. Зависимость числа итераций и скорости генерации ключа от параметра *L* представлена на рисунке 3.

Рис.3 – Зависимость числа итераций и скорости генерации ключа от параметра *L*

Таким образом, проведённое исследование показало, что эффективность синхронизации древовидных машин четности зависит от конфигурации их структурных параметров. Для систем, работающих в условиях сетевых задержек, оптимальной стратегией является использование умеренных значений *K* и *L* при максимально возможном значении *N*, которое позволяет вычислительная мощность устройств. Увеличение *N* обеспечивает кратный рост скорости генерации ключа и повышение криптостойкости, практически не увеличивая количество итераций обмена, что важно при нестабильном соединении. В то же время, чрезмерное увеличение параметра L ведет к недопустимому росту времени синхронизации.

Список использованных источников

1. Юрченков Иван Александрович, Лищенко Тимофей Викторович Синхронизация древовидных машин четности // Universum: технические науки. 2024. №5 (122).

2. Ушаков Андрей Константинович Выбор оптимальной конфигурации древовидных машин четности при их использовании для генерации секретного ключа шифрования // Научный журнал. 2018. №6

3. Урбанович, П. П. Нейросетевые технологии в криптографических приложениях : [монография] / П. П. Урбанович, М. Д. Плонковски, М. Долецки. – Минск : БГТУ, 2024. – 221 с.

УДК 681.3:553.98(574.4)

А.А. Овезова, К.Р. Аннамухаммедов, Д.М. Агаева, Е.Т. Язлыев
Международный университет нефти и газа имени Ягшыгельди Какаева,
Ашхабад, Туркменистан

РАЗРАБОТКА СИМУЛЯТОРА КИБЕРАТАКИ ЧЕРЕЗ МНОГОПОТОЧНУЮ РЕЗИДЕНТНУЮ ПРОГРАММУ

Аннотация. *Статья описывает создание симулятора кибератаки, работающего в виде многопоточной резидентной программы, имитирующей устойчивые вредоносные процессы. Представлены механизмы взаимной защиты потоков, особенности удаления и применение в обучении кибербезопасности. Симулятор позволяет формировать практические навыки противодействия атакам.*