После того, как ансамбль сетей делает вывод, составляем отчет и сохраняем. Если все 5 сверток дают не ясные результаты, то мы пробуем подать изображение еще раз. После чего можно сделать вывод. Стоит учитывать, что это – вероятность, полученная на маленьких данных, и для использования в промышленных масштабах потребуется переобучить сеть на больших данных. Это можно сделать достаточно просто, учитывая, что сеть принятия решения уже обучена, а сегментация никак не влияет на результат. Для переобучения на тот или иной рак, нужно переобучить ансамбль нейронных сетей, после чего включить блок в схему и продолжить работу. Данный подход может легко распознавать раковые заболевания точнее и быстрее человека.

Список использованных источников:
1. UNetpaper - https://arxiv.org/abs/1505.04597
2. Autodecoder - https://tensorchiefs.github.io/bbs/files/vae.pdf
3. Dropout - http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf
4. CNN - http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf
5. Biopsy - https://www.euroonco.ru/glossary-a-z/biopsy

ПРИМЕНЕНИЕ РЕЖИМА ПОНИЖЕННОГО ЭНЕРГОПОТРЕБЛЕНИЯ ДОЗУ В ИДЕНТИФИКАЦИИ ЦИФРОВЫХ СИСТЕМ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Пучков А.В.

Иванюк А. А. – д-р. техн. наук, доцент

Неотъемлемой частью любой современной цифровой системы является оперативное запоминающее устройство. Таковым нередко выступает динамическое запоминающее устройство (ДОЗУ), которое может использоваться не только по основному назначению, но и, в дополнение к этому, в качестве криптографического примитива для решения задании уникальной неклонируемой идентификации. Классическим подходом является анализ состояния ДОЗУ после включения, но не меньший интерес представляет состояние ДОЗУ после выхода из режима пониженного энергопотребления.

Структурная сложность физических, в частности, электронных систем рассматривается в физической криптографии как основа для построения криптографических примитивов. Основным понятием физической криптографии является физически неклонируемая функция (ФНФ), которую можно понимать как устройство, генерирующее значения ответов на некоторые входные воздействия (запросы), причем, пары запрос-ответ обладают уникальностью, непредсказуемостью и неклонируемостью на других экземплярах интегральных схем, выпущенных в рамках заданного технологического процесса. Процесс производства цифровых устройств и систем предполагает, что значениями отдельных их параметров принципиально невозможно управлять, задавая для них конкретные значения. Такие параметры принимают случайные, уникальные для конкретного экземпляра цифровой системы, значения, а задачей ФНФ является их извлечение и усиление. Классическим примером такого рода параметров являются задержки распространения сигналов [1]. Уникальность пар запрос-ответ означает для ФНФ возможность использовать их в рамках решения задачи неклонируемой идентификации цифровых систем, а также более сложных протоколах аутентификации на их основе. С другой стороны, случайных характер рассмотренных выше параметров позволяет использовать ФНФ для построения генераторов истинно случайных последовательностей, при этом такие генераторы сами будут являются неклонируемыми.

Для оперативных запоминающих устройств, как статических, так и динамических, характерно то, что при включении напряжения питания некоторая часть запоминающих ячеек оказывается в состоянии 1, оставшаяся часть – в состоянии 0. В самом общем случае данный процесс является случайным, а полученное распределение – уникальным со статистической точки зрения. Это даёт возможность говорить об ОЗУ как о ФНФ с ответом, представляющем собой весь массив запоминающих элементов, запросом же выступает включение питания. Поскольку чаще всего в процессе работы цифровой системы ОЗУ непрерывно используется, такой подход сопряжен со сложностями в практической реализации, т.к. использование такого криптографического примитива возможно только непосредственно после включения питающего напряжения ОЗУ, что практически всегда совпадает с включением всей цифровой системы [2]. В этой связи можно выделить ДОЗУ, для которых представляется возможным альтернативный подход – отключение регенерации части запоминающих ячеек. В отличии от предыдущего метода, это может выполняться многократно во время работы цифровой системы. В частном случае, отключение регенерации массива запоминающих элементов выполняется при переходе в режим пониженного энергопотребления, что особенно актуально для мобильных устройств. Несмотря на некоторые недостатки, из-за низкой стоимости ДОЗУ получают широкое распространение в цифровых системах различного назначения.

Для экспериментального исследования были использованы ДОЗУ Micron M45W8MW16, которыми комплектовались имеющиеся в наличии 10 плат быстрого прототипирования Digilent Nexys 4 на основе FPGA Xilinx Artix-7. Данные схемы памяти имеют объем 16 Мбайт и обладают интерфейсом, практически идентичным статическим запоминающим устройствам. При этом регенерация запоминающих ячеек

осуществляется автоматически средствами специальной логики, которой можно управлять посредством набора регистров. Был разработан аппаратно-программный комплекс, который позволяет производить операции записи и чтения из ОЗУ, устанавливать различные режимы его работы, в том числе вводить и выводить его из режима пониженного энергопотребления. Задачей экспериментов было оценить принципиальную возможность использования режима энергопотребления как альтернативного подхода для реализации ФНФ на основе ДОЗУ.

В рамках эксперимента ОЗУ было записано константными значениями '1', затем оно было введено в режим пониженного энергопотребления. Через 5 минут ОЗУ выводилось в нормальный режим работы, и его содержимое вычитывалось для анализа. Значение ’1’ было выбрано исходя из характера процессов, которые протекают в запоминающей ячейке: данное значение соответствует наличию заряда, который будет утекать из ячейки при отключении регенерации с течением времени. Чтобы выполнить первичную оценку случайности полученных данных, были построены графики частоты встречаемости для всех возможных значений, которые может принимать байт, один из которых представлен на рисунке 1. Очевидно, что распределение имеет неравномерный характер с преобладанием длинных последовательностей единичных бит.

Несмотря на это, попарное сравнение данных разных экземпляров ОЗУ показало, что они полностью различимы. Так, доля различий составила в среднем 46%, а часть карты различий для 10 устройств показана на рисунке 2.

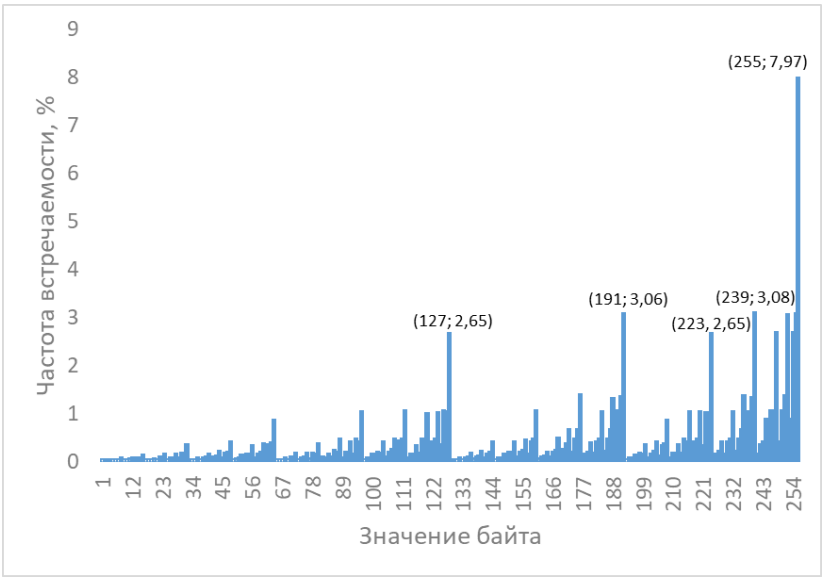

Рис. 1 – Частота встречаемости всех возможных значений байта в ДОЗУ после выхода из режима пониженного энергопотребления

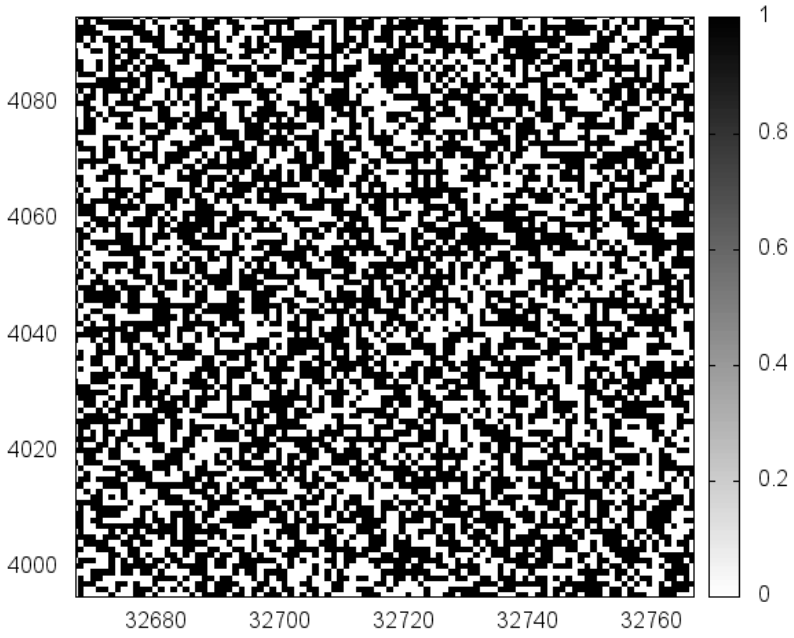

Рис. 2 – Часть карты различий состояний запоминающих ячеек для 10 экземпляров ДОЗУ

Таким образом, в ходе первичных экспериментов было показано, что с использованием режима пониженного энергопотребления можно осуществлять извлечение уникальных характеристик ДОЗУ, что является альтернативным подходом, не требующим выключения и включения всей цифровой системы. Кроме того, использование режима пониженного энергопотребления представляет интерес для мобильных устройств. Однако первичные данные говорят о низкой случайности данных, полученных данным методом, что не дает использовать его для построения генераторов истинно случайных последовательностей.

Список использованных источников:

1. Tao, S. Reliable low-overhead arbiter-based physical unclonable functions for resource-constrained IoT devices / S. Tao, E. Dubrova // Proc. of the 4th Workshop on Cryptography and Security in Computing Systems. – Stockholm, Sweden, 2017. – P. 1–6.

2. Rosenblatt, S. A Self-Authenticating Chip Architecture Using an Intrinsic Fingerprint of Embedded DRAM / S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, S. S. Iyer // IEEE Journal of Solid-State Circuits. – 2013. – Vol. 48, No. 11. – P. 2934-2943.

KUBERNETES КЛАСТЕР ДЛЯ ПОЛНОТЕКСТОВОГО ПОИСКА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сенькович Д.С.

Жвакина А. В. – к. т. н, доцент

На сегодняшний день в разработке программного обеспечения используется все больше независимых взаимодействующих между собой приложений. Особенно остро проблема проявляется при разработке приложений по обработке больших данных. Целью проводимых исследований является создание кластера Docker контейнеров с Apache Hadoop и Apache Solr под управлением Kubernetes. Решение полезно как для обучения технологии полнотекстового поиска, так и при развертывании кластера для тестирования и разработки.

Задача поиска по тексту предполагает работу со сложными индексами и большими объемами данных. Поэтому для ее решения образовалась целая группа отдельных инструментов.

Многие базы данных имеют встроенную возможность поиска по тексту, однако обычно это очень ограниченная реализация. В большинстве случаев следует использовать более подходящие технологии. Как следствие, получили широкое распространение технологии полнотекстового поиска, например, Apache Solr и Elasticsearch.

Solr − платформа полнотекстового поиска с открытым исходным кодом, основанная на проекте Apache Lucene. Ее основные возможности: полнотекстовый поиск, подсветка результатов, фасетный поиск, динамическая кластеризация, интеграция с базами данных, обработка документов со сложным форматом (например, Word, PDF). Так как в Solr есть возможность распределенного поиска и репликации, Solr хорошо масштабируем. По состоянию на май 2016 года Solr является вторым по популярности поисковым движком, уступая Elasticsearch.

Solr и Elasticsearch являются лидерами на рынке полнотекстового поиска. Более популярным решением является Elasticsearch ввиду простоты начала работы с ним. Solr в то же время имеет большую историю и является решением с открытым исходным кодов, также как и его экосистема. Обе системы богаты своими возможностями и не уступают друг другу по скорости, используя один движок для полнотекстового поиска Apache Lucene. Принимая во внимание, что и движок, и Apache Solr разрабатываются одной команды, можно предполагать лучшую интеграцию и использование возможностей движка. Из различий также можно отметить более долговременную поддержку существующих возможностей в Apache Solr и возможность использовать встроенное машинное обучение, что возможно только в платной версии Elasticsearch.

Solr использует Apache HDFS для хранения документов и индексов, что обеспечивает сохранность и быстрый доступ.

HDFS (Hadoop Distributed File System) -- файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера. Все блоки в HDFS (кроме последнего блока файла) имеют одинаковый размер, и каждый блок может быть размещён на нескольких узлах, размер блока и коэффициент репликации (количество узлов, на которых должен быть размещён каждый блок) определяются в настройках на уровне файла. Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов. Файлы в HDFS могут быть записаны лишь однажды (модификация не поддерживается), а запись в файл в одно время может вести только один процесс. Организация файлов в пространстве имён − традиционная иерархическая: есть корневой каталог, поддерживается вложение каталогов, в одном каталоге могут располагаться и файлы, и другие каталоги.

В связи с увеличением количества компонентов (программ) при разработке программного обеспечения установка и мониторинг, подготовка среды для предложения становятся сложнее. Поэтому активно развиваются технологии виртуализации, в частности, контейнеризации - виртуализации уровня ОС. Наиболее популярным решением является Docker.

Docker − это инструмент, предназначенный для упрощения создания, развертывания и запуска приложений с помощью контейнеров. Контейнеры позволяют разработчику упаковать приложение со всеми необходимыми ему компонентами, такими как библиотеки и другие зависимости, и поставлять все это как один пакет. Таким образом, благодаря Docker, разработчик может быть уверен, что приложение будет