ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ ПРОГРАММНЫХ ПЛАТФОРМ NODE.JS, PYTHON И GO

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

Марчук М. С., Саскевич А. В., Цыркунов Д. А.

Стройникова Е. Д. – ассистент кафедры информатики

В данной работе исследованы криптографические возможности программных платформ Node.js, Python, Go по следующим критериям: 1) основные особенности программных платформ, причины их популярности, тенденции в развитии; 2) криптографические возможности, наличие собственных и сторонних библиотек шифрования и хеширования, их популярность и поддержка; 3) сравнительная характеристика основных криптографических алгоритмов, реализованных на данных платформах; 4) особенности применения на практике.

Основные особенности программных платформ, причины их популярности, тенденции в развитии.

Node.js – на данный момент одна из наиболее популярных программных платформ в сфере веб-технологий. Благодаря четко сформулированной идеологии, простоте, переносимости и достаточно быстрой скорости работы платформа нашла свое применение у таких крупных компаний, как Ebay, PayPal (банковские системы), Wikipedia (online-базы знаний), Mozilla и Yahoo (информационные технологии), New York Times и некоторых других СМИ и пр. В последние годы Node.js набирает все большую популярность и пользуется широкой поддержкой среди разработчиков, которые предлагают множество готовых реализаций различных видов алгоритмов и программных продуктов. Данные решения любой желающий может применить в своей работе.

Платформа Python занимает почетное место среди других программных платформ. Особенностями одноименного языка программирования являются простота синтаксиса и наличие множества библиотек. Python зарекомендовал себя в основном как платформа для веб-разработки и машинного обучения, однако он применим и для более широкого круга задач. Список компаний, которые используют Python, длинный. Среди них Google, Facebook, Yahoo, NASA, Red Hat, IBM, Instagram, Dropbox, Pinterest, Quora, Яндекс, Mail.Ru. Но еще больший список можно составить из программного обеспечения, написанного с применением Python. Это BitTorrent, GIMP, World of Tanks, Battlefield 2, Blender, Sublime Text и пр. Стремительно растущее количество Python-специалистов говорит о том, что программная платформа, действительно, может быть применена в самых различных областях промышленности.

В отличие от вышеописанных платформ, которые основаны на интерпретируемых языках, платформа Go реализует компилируемый язык программирования, который по скорости работы приближается к таким языкам, как C и C++. Кроме того, платформа реализует экосистему управления модулями и библиотеками, близкую к экосистемам npm (Node.js) и pip (Python). В последнее время Go также набирает популярность, однако на данный момент популярность Go еще не соизмерима с популярностью Node.js и Python. Поддерживаемая командой разработчиков Google, платформа применяется такими компаниями, как Facebook, Dropbox, Google, GitHub, IBM, Twitter (информационные технологии) и пр.

Криптографические возможности программных платформ, наличие собственных и сторонних библиотек шифрования и хеширования, их популярность и поддержка.

Для работы с алгоритмами шифрования и хеширования платформа Node.js предлагает модуль crypto, оснащенный довольно большим функционалом, реализущим такие алгоритмы, как DHE, EDH, AES, RC4, RC2, MD5, SHA256, SHA384, kGOST, RSA и др. У данного модуля есть популярный аналог crypto-js, реализующий алгоритмы SHA256, AES, MD5, RC4 и др. Помимо этого, есть еще несколько менее популярных криптографических модулей.

Платформа Python «из коробки» предлагает модули hashlib и hmac, предназначенные для вычисления хешей и цифровых подписей. Данные модули реализуют такие алгоритмы шифрования, как SHA1, SHA224, SHA256, SHA384, SHA512 и MD5. Кроме того, в модуле zlib присутствуют алгоритмы CRC32 и ADLER32. Из популярных сторонних библиотек можно отметить модули pycrypto (реализует криптографические алгоритмы шифрования AES, ARC2, ARC4, Blowfish, DES, DES3, хеш-функции CMAC, HMAC, MD5, SHA, SHA1, RIPEMD и дополнительно несколько алгоритмов цифровой подписи) и cryptography (реализует стандарты X.509, хеш-функции SHA, MD5, RIPEMD, Whirlpool, алгоритмы шифрования Blowfish, ARC4, RSA и DSA).

Платформа Go также имеет богатый функционал для хеш-функций (MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512, RIPEMD и др.) и алгоритмов шифрования (RC4, RSA, AES, DES, DSA). Данного набора хватает для большинства нужд, потому сторонние крупные и популярные библиотеки, реализующие дополнительный функционал, отсутствуют.

Сравнительная характеристика основных криптографических алгоритмов, реализованных на программных платформах.

Для сравнения были выбраны наиболее популярные и известные алгоритмы хеширования MD5, SHA256, SHA512 и алгоритмы цифровой подписи и шифрования RC4, RSA, DSA. В качестве основных критериев для сравнения алгоритмов выбраны скорость работы, основные параметры работы (разрядность, криптостойкость, тип). В табл. 1 и табл. 2 ниже представлены сравнительные характеристики скорости работы алгоритмов с использованием процессора Intel Core i7 2.4MHz 4GB RAM.

Таблица 1. Среднее время хеширования случайных данных размером 512кб

Алгоритм	Node.js	Python	Go
MD5	2.23 мс	0.98 мс	0.75 мс
SHA256	2.39 мс	1.82 мс	0.38 мс
SHA512	2.12 мс	2.14 мс	0.49 мс

Таблица 2. Среднее время генерации случайного ключа размером 1024 бита

Алгоритм	Node.js	Python	Go
RSA	320.5 мс	580.5 мс	60.7 мс
DSA	N/A	674.8 мс	N/A

Особенности применения криптографических алгоритмов на практике.

В соответствии с результатами исследования, при выборе конкретного алгоритма следует в первую очередь оценить, насколько критичными являются скорость и уровень защищенности алгоритма для данной задачи. Наилучшим вариантом является применение алгоритмов, реализованных на уровне ОС или оборудования. Помимо этого, следует обратить внимание на то, что гораздо безопаснее применять алгоритмы, реализованные «из коробки», – таким образом можно минимизировать вероятность появления ошибок и увеличить устойчивость к попыткам взлома.

Список использованных источников:

1. Node.js Crypto Documentation [Electronic resource] / Node.js v5.9.1 Documentation. – Node.js Foundation, 2016. – Mode of access: https://nodejs.org/api/crypto.html. – Date of access: 26.03.2016.
2. go/crypto documentation [Electronic resource] / The Go Programming Language. – Build version go1.6, 2016. – Mode of access: https://golang.org/pkg/crypto/. – Date of access: 27.03.2016.
3. Python hashlib documentation [Electronic resources] / Python. – Python Software Foundation, 2016. – Mode of access: https://docs.python.org/3/library/hashlib.html. – Date of access: 26.03.2016.
4. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.

СОБЫТИЙНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ОБРАБОТКИ НАБОРОВ ДАННЫХ

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

Сафонов А. А.

Ганжа В. А. – кандидат физико-математических наук, доцент

С ростом объёмов информации, требуются новые способы их обработки. Пользователям требуется быстрый доступ к новой информации и возможность получения данных в реальном времени. Это стимулирует разработчиков создавать отзывчивые интерфейсы и модели для обработки.

Разработанная модель позволяет создавать и преобразовывать наборы данных и реагировать на их изменения. Она направлена на повышение качества кода при написании решений для обработки данных. В текущей реализации только на платформе .NET Framework.

На рисунке 1 представлена схема обработки запроса на вставку в набор чисел: