Protecting Nuclear Safeguards Monitoring Data from Tampering
Roger G. Johnson, Michael J. Timmons. John S. Warner
Science and Global Security, 2007*,* Volume 15, pp. 185-209

ЗАЩИТА МОНИТОРИНГА ЯДЕРНЫХ ГАРАНТИЙ ОТ ВЗЛОМА ДАННЫХ

Роджер Дж. Джонстон, Майкл Дж. Тиммонс и Джо С. Уорнер

Эффективный мониторинг договора нуждается в надежности информации ядерного мониторинга от попыток взлома. Но пломбы с индикацией взлома и стандартные методики зашифровки (или целостности) данных не обеспечивают надежной безопасности, особенно против электронных и физических вторжений в содержимое международных ядерных гарантий. В этой статье представлен альтернативный подход для гарантии целостности контролируемых данных, называемый «Одноразовый блокнот для подстановки знаков» (ОБДПЗ). Тот шифр служит сочетанием непробиваемой одноразовой клавиатуры и традиционного подстановочного шифра. ОБДПЗ обеспечивает непробиваемую безопасность до вторжения противника в аппаратуру ядерного мониторинга (даже если вторжение пройдет незамеченным) и хорошую безопасность в дальнейшем.

Роджер Дж. Джонстон и Джо С. Уорнер работают в Отделении ядерной техники Аргоннской национальной лаборатории, Аргонн, Иллинойс, США, Майкл Дж. Тиммонс работает в Университете Св. Эндрюса, Сент-Эндрюс, Файф, Шотландия.

Статья получена редакцией 5 марта 2007 г. и принята к публикации 24 июля 2007 г.

Эта работа была выполнена под покровительством Министерства энергетики США (МЭ), LAUR-07-0884. Высказанные в ней взгляды принадлежат авторам и не должны обязательно быть приписаны Лос Аламоской национальной лаборатории, Аргоннской национальной лаборатории, МЭ или правительству США.

Почтовый адрес для корреспонденций: Roger G. Johnston, Ph.D., Vulnerability Assessment Team, Nuclear Engineering Division, Argonne National Laboratory, 9700 South Case Avenue, Building 206, Argonne, Illinois, 60439-4825, USA. E-mail: rogerj@anl.gov

ВВЕДЕНИЕ

Международные ядерные гарантии («мониторинг договора») – это весьма необычный тип приложений безопасности. [1] В отличие от обычных приложений безопасности (например, *внутренних* ядерных гарантий) противник в международных гарантиях (то есть, «хозяин», или страна, подвергающаяся контролю) владеет имуществом и установками, представляющими интерес. Противник по причинам надежности, безопасности, контрразведки, взаимности, национализма и геополитики будет часто настаивать на полном и детальном понимании стратегии мониторинга, аппаратуры и методики безопасности, используемых инспекторами. Напротив, при обычных приложениях безопасности противников обычно не инструктируют Другие стороны международных ядерных гарантий также весьма уникальны в сравнении с внутренними ядерными гарантиями и другими более шаблонными видами приложений безопасности.[1]

Надежная проверка договора особенно нуждается в том, чтобы инспекторы могли доверять правдивости данных мониторинга или обзора, собранных на ядерной установке (или около нее) при помощи контролирующей аппаратуры, которая обычно не посещается после запуска. Такая аппаратура может состоять, например, из сейсмометров, радиологических измерительных приборов, видео- или фото-систем обзора, устройств контроля доступа, счетчиков автомашин и детекторов вторжения. Впрочем, как может такая информация остаться безопасной, особенно если противник в случае международных ядерных гарантий имеет полный доступ к технически деталям аппаратуры мониторинга, не считая ресурсов национального или мирового уровня и технической экспертизы, которые потенциально можно использовать для обмана?

Традиционно применяются пломбы с индикацией взлома [2-4] для обнаружения открытия или взлома аппаратуры, электроники или шкафов с инструментами, а стандартные методики зашифровки данных (или их целостности) используются для защиты записанной или пере-

данной информации. К сожалению, как будет обсуждаться позднее, эти подходы не целиком надежны для международных ядерных гарантий.

В этой статье представлена новая методика (включающая комбинацию двух ранее не связанных старых методик) для гарантированной достоверности данных, названная ОБДПЗ (Одноразовый блокнот для подстановки знаков). ОБДПЗ оказался простой, быстрой и высоко безопасной методикой для сохранения записываемых или передаваемых данных. [5] Она практична с вычислительной точки зрения для недорогих микропроцессоров. ОБДПЗ не имеет патентных или лицензионных вопросов, а также проблем с экспортным контролем (в отличие от многих других методик шифрования и идентификации) и хорошо подходит для применения в международных ядерных гарантиях в добавление к некоторым другим видам приложений регистрации данных, где критична безопасность, а противник все еще способен понимать детали аппаратуры мониторинга и может исподтишка получить физический доступ к ней. Примером может быть регистратор информации о положении в системе глобального позиционирования. [6]

ПОЧЕМУ НЕ ПОДХОДЯТ ПЛОМБЫ?

Группа по оценке уязвимости (ГОУ), которая раньше находилась в Лос Аламосской национальной лаборатории, пространно рассмотрела сотни различных пломб с индикацией взлома за последние 15 лет. ГОУ показала, что все изученные пломбы можно было быстро разрушить, используя только простейшие инструменты, подводы и методы, доступные почти любому [2, 7-9] (понятие «разрушить» пломбу означает удалить ее, а после кражи ли взлома содержания контейнера использовать для опечатывания оригинальную пломбу или поделку, причем все сделать без возможности обнаружения). ГОУ еще не встречала пломбу, включая пассивные или электронные пломбы, применявшиеся для ядерных гарантий, которой требовались искушенные противник или нападение, чтобы разрушить ее.

Существуют практические меры противодействия для большинства продемонстрированных разрушений пломб. Обычно от пользователя пломбой требуется усовершенствовать способ установки и проверки пломбы. Необходима также интенсивная тренировка рук для установщиков пломб и инспекторов, чтобы были понятны уязвимости конкретных пломб, которыми они пользуются, и известно, как выглядят наиболее вероятные сценарии нападения. К несчастью, все это требует больше времени, денег и усилий, чем большинство программ безопасности хотели бы вложить даже для внутренних или международных ядерных гарантий.

К счастью, возможны лучшие пломбы.[2, 10] Обычные пломбы можно видоизменить для затруднения нападений. Еще более эффективный вариант заключается в использовании фундаментально нового подхода к обнаружению взлома, названного авторами «анти-информационными» пломбами (АИП). [2, 10] Обычные пломбы также часто способны обнаружить взлом достаточно хорошо, но должны сохранить эту информацию на пломбе (или внутри ее) до того, как пломбу можно будет проинспектировать. Но противник также легко сможет скрыть или стереть это «состояние тревоги» или же заменить пломбу свежей подделкой.

В случае АИП информация закладывается для хранения на пломбе (ими внутри ее) с момента ее установки, и это пока указывает на отсутствие вскрытия. Когда вскрытие обнаружено, эта секретная «анти-информация» (обычно байт или два) внезапно стирается. [11] Отсутствие «анти-информации» во время проверки печати указывает на то, что вскрытие произошло. При таком подходе противник не может спрятать или стереть «состояние тревоги»; фальсификация опечатывання пломбы не даст ему ничего, если он не знает, какую «анти-информацию» надо занести в пломбу или поместить на ней.[12]

Авторы считают, что АИП смогут обеспечить гораздо более надежное обнаружение взлома, нежели обычные пломбы. Как показывает опыт автора, однако, существует относительно мало серьезного интереса в любом месте к более лучшим пломбам с индикацией взлома, включая приложения к ядерным гарантиям (внутренним или международным). Более того, маловероятно, чтобы АИП, даже если они окажутся лучше обычных пломб, смогли обеспечить абсолютные гарантии относительно обнаружения взлома. Следует продолжить применение пломб, но они не служат панацеей для уверенности, что данные мониторинга (или оборудование) свободны от взлома.[13]

ПОЧЕМУ НЕ ПОДХОДЯТ ОБЫЧНЫЕ МЕТОДИКИ ДЛЯ ЦЕЛОСТНОСТИ ДАННЫХ?

Вызывающи тревогу критический вопрос в этой статье относится к целостности данных по мониторингу, собранных для проверки согласия страны с ядерными договорами, соглашениями и международными обязательствами. [14] Целостность данных обычно проверяется при использовании «мусора» (hashes), зашифрованного или нет, или же близко связанных с ними кодами идентификации послания (КИП). [15-17] «Мусор» - это числа с фиксированной длиной, выбранные из более крупного набора данных, которые обычно заметно изменяются, когда меняются все данные или часть данных из оригинального набора. Проверка суммы всех цифр служит примером простого мусора.

Полная зашифровка данных (с использованием шифра) также может применяться (хотя это необычно) для проверки целостности данных в ситуациях, где противник может пожелать сделать конкретные несанкционированные изменения в данных, чтобы скрыть факты обмана. [18] Шифры обычнее применяются в наше время к приложениям с конфиденциальностью данных, где кто-то хочет передавать данные между двумя *физически безопасными* местами [19], так что любой, перехватывающий зашифрованную информацию, не сможет понять, что она значит. Оригинальные секретные данные (или послания) обычно называются «чистым текстом», а зашифрованные данные (или послания) называются «шифрованным текстом». Для большинства приложений, содержащих применение шифров для обеспечения конфиденциальности данных, считается, что «плохие парни» знают шифрованный текст и алгоритм шифровки, но не чистый текст или секретный ключ для расшифровки. Это не так для международных ядерных гарантий, где противник будет обычно знать чистый текст по причинам, обсуждаемым позднее, и где конфиденциальность данных нежелательна по причинам прозрачности.

Одна из причин рассматривать использование шифра для целостности данных (хотя это и необычно), заключается в том, что обычно более трудно «пробить» усложненный шифр, то есть, вычислить секретный ключ (даже зная чистый текст), чем создать новый документ с теми же самыми «мусором» и КИПами, как ц оригинала. Более того, уровень усилий, требуемых для «пробития» шифра, обычно хорошо понимается, а безопасность, обеспечиваемая «мусором» или КИП, часто остается неясной. [20]

Существуют четыре основные проблемы при использовании «мусора», КИПов или обычных шифров для гарантирования правдивости данных ядерного мониторинга. Во-первых, современные методики, которые опираются на секретные ключи (будь то «мусор», КИПы или шифры) являются безопасными только с вычислительной точки зрения, а не абсолютно безопасными. [15-17, 21]. Это означает следующее: эксперты *думают*, что вычисление секретного ключа или способность создать другими способами фальшивые данные, которые кажутся идентичными, должно потребовать громадных математических и вычислительных ресурсов. Проблема международных ядерных гарантий, конечно, в том, что ядерный противник (это целая страна) обычно будет иметь значительные ресурсы, включая математиков и криптоаналитиков мирового класса и компьютеры или, по крайней мер, доступ к ним. Более того, как напоминает история. «мусор» (зашифрованный или нет), КИПы и шифры, которые когда-то казались безопасными с вычислительной точки зрения, оказывались раскрытыми (иногда неожиданно легко) по мере того, как появлялись новые методики криптоанализа, новая вычислительная мощность и новая экспертиза. [22]

Вторая серьезная проблема с «мусором», КИПами или обычными шифрами состоит в том, что они не обеспечивают значительной безопасности, если противник сможет получить доступ к местам отправления или приема информации. Это позволит ему получить применяемые секретный код (коды) и/или алгоритмы, а также непосредственно вскрыть открытый текст и этим самым заменить реальную информацию своей фальшивой. Теоретически, если вторжение в аппаратуру мониторинга может быть надежно обнаружено, а ключ (ключи) шифровки стерты достаточно быстро, у противника появятся более трудные проблемы с фальсификацией данных мониторинга (хотя теоретически это еще возможно). Однако, пломбы и современные методы обнаружения взлома не подходят для такой угрозы, как обсуждалось ранее. Кроме того, время на полное стирание ключей к современным шрифтам может оказаться относительно долгим на микропроцессорах, так как минимально рекомендуемый размер ключа в приложениях с высокой безопасностью составляет 2048 бит, или 256 байтов. [23,24] Ключи для КИПов имеют тенденцию быть меньше, но все еще требуют ми-

нимально 128 байтов, чтобы быстро и надежно стираться, а часто больше при высоком уровне безопасности.

Третья проблема с обычными методами связана с тем, что даже если секретный ключ (ключи) полностью стерты до того, как противник еще может восстановить его, противник в случае международных ядерных гарантий обычно будет знать чистый текст, шифрованный текст и используемый шифр или алгоритм идентификации данных. [25] Он будет знать чистый текст, так как именно на его установке производятся измерения мониторинга, и ему будет понятно, что происходит на его собственной аппаратуре. Он также сможет получить шифрованный текст (хранимые зарегистрированные данные) и в случае необходимости алгоритм расшифровки/целостности данных путем вторжения в электронику или проникновения другими способами в микропроцессор. Фактически он может автоматически узнать алгоритм, поскольку вероятно от инспекторов будут требовать раскрыть его по причине прозрачности и поскольку проверяемая страна может настаивать на том, чтобы ей был представлен исходный код математического обеспечения. [26] В этом случае гораздо легче раскрыть шифр (то есть, вычислить секретный код), если противнику известны чистый текст, шифрованный текст и алгоритм шифра.[27] Такая ситуация нетипична для обычных приложений шифрования (хотя обычна для приложений «мусора» и КИП).

Четвертая проблема с «мусором», КИПами и обычными шифрами заключается в том, что они требуют много вычислений и поэтому их трудно эффективно осуществить на микропроцессоре [28], что делает их непрактичными для небольшого, экономичного, прозрачного оборудования мониторинга в рабочих условиях. [29]

ОДНОРАЗОВАЯ КЛАВИАТУРА

Существует только один шифр, который, как доказано математически, нельзя раскрыть на все времена, - это одноразовая клавиатура. [21, 30] Этот шифр, известный также как одноразовый блокнот (ОБ) или шифр Вернана, был изобретен примерно в 1917 году Кроме своей нераскрываемости ОБ обладает теми преимуществами, что не имеет патентных и лицензионных вопросов, а также проблем экспортного контроля (в отличие от многих современных шифров и КИПов), а также прост и очень быстр.

Идея этого шифра в том, чтобы использовать полностью случайный ключ той же длины, что и чистый текст. Этот ключ можно использовать только один раз, а затем он должен быть выброшен. [31] Хотя советские шпионы широко пользовались одноразовой клавиатурой в прошлом столетии, она не считалась практичной для многих приложений из-за больших требований к памяти для ключа. Однако, этот недостаток становится менее важным с учетом уменьшения цен и размера средств цифровой памяти. ОБ, как правило, применяется для шифровки буквенных знаков, но сейчас мы продемонстрируем его использование для шифрования числовых значений, которые больше подходят к информации мониторинга. Допустим, что кто-то захотел зашифровать цифры «1663», а первые четыре случайные цифры в одноразовом блокноте равны «3907». Тогда имеем

Чистый текст	1	6	6	3
ОБ	3	9	0	7

Складываем численные значения чистого текста и ОБ и получаем

Сумма: чистый текст + ОБ	4	15	6	10

Значения суммы, превышающие 9, «свертываем» путем вычитания 10, то есть, проводим расчет по модулю 10.33 и получаем в результате

Сумма по модулю 10	4	5	6	0

Зашифрованное послание (шифрованный текст) поэтому выглядит как «4560». Чтобы расшифровать его, ведем процесс в обратном направлении путем вычитания значений ОБ с применением модуля 10.

Обратите внимание, что одного только шифра ОБ недостаточно, чтобы создать безо-

пасность для данных мониторинга гарантий по двум причинам. Во-первых, для микропроцессора трудно быстро стереть большой одноразовый блокнот (или шифрованный текст), как только обнаружится физическое или электронное вторжение. Во-вторых, даже если ОБ стерся, противник может тривиально восстановить его, потому что он знает чистый текст и может прочитать хранимый шифрованный текст. Восстановление ОБ позволит ему заменить истинные данные мониторинга на поддельные данные, которые будут казаться достоверными инспекторам.[34]

ПОДСТАНОВОЧНЫЙ ШИФР

Даже более простым, чем одноразовый блокнот, является подстановочный шифр [17, 35], которому тысячи лет и поэтому у него нет патентных и лицензионных вопросов, а также проблем экспортного контроля. Этот шифр, впрочем, может быть легко раскрыт даже любителями. Как подсказывает название, подстановочный шифр предполагает замену каждого знака чистого текста на другой, заранее определенный знак шифрованного текста. Например, предположим, то подстановочный шифр задается следующими однозначными связями между каждой десятичной цифрой чистого текста (верхняя строка) и расположенной прямо снизу от нее десятичной цифрой в нижней строке:

чистый текст	0 1 2 3 4 5 6 7 8 9
отображение	2 9 8 3 7 1 0 6 5 4

Секретный нижний список цифр, приведенных в случайном порядке (причем ни одна цифра не повторяется) определяет подстановочный шифр и называется ключом или «отображением», поскольку он определяет, как цифры чистого текста отображаются в цифры шифрованного текста. Так, в приведенном выше примере 1 отображается в 9, 6 отображается в 0, 3 отображается в 3 и так далее, Поэтому, например, чистый текст «1663» шифруется в шифрованный текст «9003». Для дешифровки отображение (ключ) используется в обратном порядке.

В традиционном подстановочном шифре отображение не меняется вне зависимости от длины послания. Данная цифра чистого текста всегда будет зашифрована в одну и ту же цифру шифрованного текста независимо от того, где эта цифра появляется в чистом тексте. [36] Вот это и делает безопасность подстановочного шифра столь низкой в сравнении с ОБ, где (аддитивный) ключ всегда меняется. [37] Как и ОБ, обычный подстановочный шифр бесполезен сам по себе для защиты данных мониторинга, особенно когда противник может проникнуть в аппаратуру мониторинга без надежного обнаружения.

ОДНОРАЗОВЫЙ БЛОКНОТ ДЛЯ ПОДСТАНОВКИ ЗНАКОВ (ОБДПЗ)

ОБДПЗ – это метод шифрования, который объединяет одноразовый блокнот и подстановочный шифр. Его можно представить как подстановочный шифр, где случайное отображение (или «ключ») меняется для каждого знака или цифры, подлежащих зашифровке. Этап шифрования происходит тем же образом, как в подстановочном шифре; единственная разница в том, что изображение меняется после каждой зашифровки цифры чистого текста, а каждое изображение после использования немедленно стирается. Можно думать и по-другому: ОБДПЗ служит одноразовым блокнотом для подстановки отображений и каждое из них отбрасывается после того, как применяется только однажды для зашифровки одного знака или цифры.[38]

Рассмотрим следующий пример, где шифруются данные чистого текста «1663». Первое отображение, приведенное ниже, используется для подстановки «2» в качестве первой цифры («1») чистого текста; теперь первой цифрой шифрованного текста поэтому становится «2». Чтобы найти подстановку для второй цифры, переходим к следующему случайному отображению после стирания первого. Это значит, что «6» в чистом тексте заменяется на «8». Такой процесс повторяется до тех пор, пока не будет зашифрован весь чистый текст.

Рассмотрим отображение для первой цифры чистого текста.

цифра чистого текста	0 1 2 3 4 5 6 7 8 9

отображение	9 2 4 1 7 3 6 5 0 8

Так что, например, «1» переходит в «2».

цифра чистого текста	0 1 2 3 4 5 6 7 8 9
отображение	2 4 0 1 5 7 8 6 9 3

Так что, например. «6» переходит в «8».
Строим отображение для третьей цифры чистого текста.

цифра чистого текста	0 1 2 3 4 5 6 7 8 9
отображение	9 2 1 7 5 3 0 6 8 4

Так что, например, «6» переходит в «0».
Строим отображение для четвертой цифры чистого текста.

цифра чистого текста	0 1 2 3 4 5 6 7 8 9
отображение	1 2 6 7 0 8 5 9 3 4

Так что, например, «3» переходит в «7».

Таким образом, эти отображения дают нам шифрованный текст «2807».

Этот метод ОБДПЗ столь же силен (нераскрываем), как обычный одноразовый блокнот (ОБ) только при нападении на шифрованный текст. Но в отличие от обычного ОБ, если одновременно известны чистый и шифрованный тексты, противник все еще не может восстановить использованные и стертые отображения, даже если доступны неограниченные количества чистого текста и шифрованного текста. Это потому, что единственной информацией , которая может быть конфиденциально заключена там, оказывается величина всего лишь одной из десяти цифр в каждом ранее использованном отображении. Поскольку противник хочет зашифровать различный набор цифр чистых текстов, которые фактически зашифрованы, он застревает. [39] Не имея никаких подсказок о том, какими были использованные отображения., он имеет всего один шанс из девяти [40] догадаться о правильной цифре шифрованного текста, соответствующей каждой цифре чистого текста, который он хотел бы фальсифицировать.

Сейчас в данные мониторинга ядерных гарантий будут, как правило, включаться количественные показания датчика, записанные микропроцессором. Поэтому имеет больше смысла использовать гексадесятичные («гексы», или на базе 16) цифры, а не десятичные (база 10) цифры для чистого текста, шифрованного текста и отображений ОБДПЗ. [41] При использовании гекс-цифр каждое отображение ОБДПЗ состоит из 16 гексов, расположенных в случайном порядке, причем ни одна из гекс-цифр не повторяется в данном отображении.

Каждое из гекс-отображений ОБДПЗ поэтому будет случайной перестановкой набора гекс-цифр 0, 1, 2, 3, 4, 5, 6. 7, 8, 9, A, B, C, D, E, F. Существует свыше 20 триллионов (16!) различных возможных отображений. В результате совпадения одно отображение может оказаться идентичным другому, но только если это повторяется непредсказуемо.

БЕЗОПАСНОСТЬ ДАННЫХ ОБДПЗ

Теперь с ОБДПЗ, даже если вторжение противника пройдет незамеченным, он не сможет подделать данные, полученные до его вмешательства (считая, что отображения, действительно, стерты безоговорочно после того, как они использованы). Нет необходимости никакого стирания данных, чтобы защитить правдивость записанных ранее данных по мониторингу. [42]

Но что по поводу фальсификации будущей информации? Если только все *неиспользованные* отображения нельзя будет стереть мгновенно, когда обнаружится вторжение (а это, как обсуждалось ранее, не может быть гарантированно), тогда отображения ОБДПЗ будут доступны противнику для фальсификации будущей информации.

Путь к решению этой проблемы заключается в том, чтобы микропроцессор отбирал каждое отображение, когда в нем возникает необходимость, из кэш-памяти неиспользованных

отображений таким способом, который известен только инспекторам. Наиболее просто это может быть сделано с помощью генератора псевдослучайных чисел (ГПСЧ) – простого итерационного уравнения, которое установленным и повторяющимся образом создает псевдослучайные числа из ранее созданных чисел. Новое псевдослучайное число указывает, какое отображение должно использоваться следующим. ГПСЧ запускается секретным ключом (начальным числом), известным только инспекторам, хотя не требуется, чтобы алгоритм ГПСЧ сам по себе был секретным. Общей формой для ГПСЧ служит линейный конгруентный генератор, создающий псевдослучайную последовательность целых чисел (I_0, I_1, I_2,….) в интервале [0, M-1] по формуле $I_{n+1}= (AI_n + C) \bmod M$, где целочисленные коэффициенты A, C и M должны быть тщательно выбраны, а I0 является ключом. [43-45]

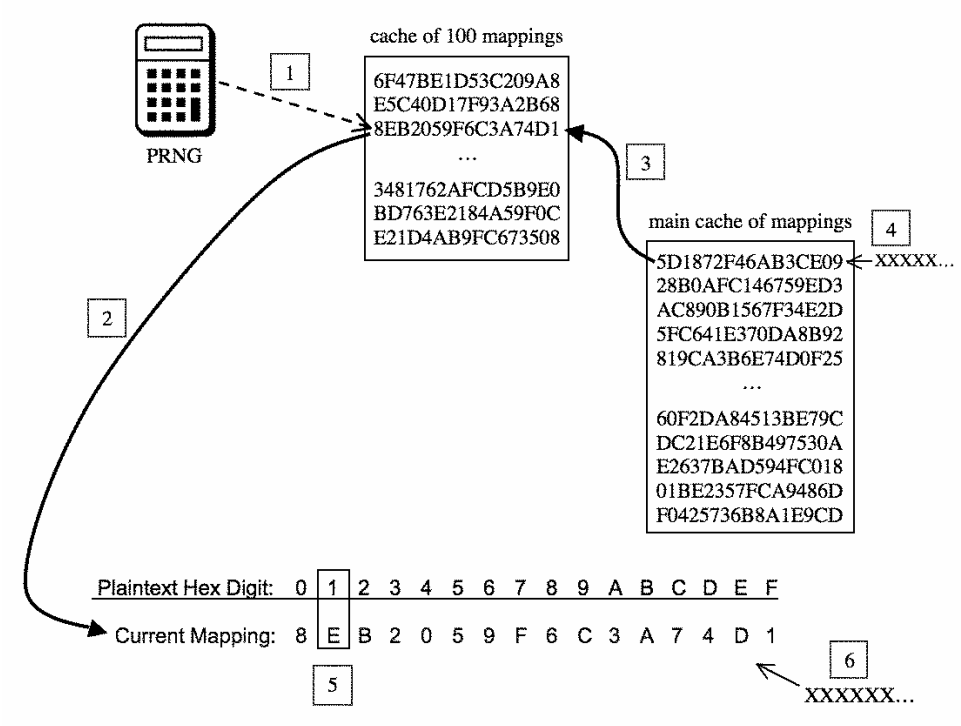

Рис. 1: Схема алгоритма ОБДПЗ. На первом этапе ГПСЧ (в левом верхнем углу) выбирает одно из отображений кэш-памяти, содержащей 100 случайных отображений. На втором этапе отображение переносится к месту, где оно может быть использовано в качестве текущего отображения для шифрования ОБДПЗ одной гекс-цифры. Перемещенное отображение немедленно замещается в кэш-памяти на 100 мест новым доступным отображением из основной кэш-памяти (третий этап). Это отображение, в свою очередь, немедленно и бесповоротно стирается (знак стирания – xxxxx) из основной кэш-памяти (четвертый этап). На пятом этапе текущее отображение использовано для изменения одной гекс-цифры чистого текста (1 в этом примере) в одну гекс-цифру шифрованно текста (E). Наконец, текущее отображение стирается (шестой этап). К концу этапа 6 полезная информация о текущем отображении постоянно потеряна для противника. Если обнаружено вторжение, текущее итеративное значение ГПСЧ (обычно длиной в два байта) немедленно стирается, оставляя противника без указаний на то, какое из 100 отображений в малой кэш-памяти намечается к использованию следующим. Для еще более лучших шансов (для хороших парней) больше сотни отображений может храниться в меньшей кэш-памяти.

На Рис.1 показано, как этот процесс отбора мог бы осуществиться наиболее эффективно. ГПСЧ отбирает отображение из кэш-памяти на, скажем, 100 отображений ОБДПЗ. После того, как каждое отображение используется, оно заменяется новым доступным отображением из большей кэш-памяти отображений. ГПЧС создает двухбайтовое число 0-65535. Свертывание этого числа по модулю 100 указывает, какое из 100 отображений меньшей кэш-памяти будет использовано следующим. Если обнаружится вторжение, стирание текущего двухбайтового значения ГПСЧ оставляет взломщика без намека на порядок, в котором

должны выбираться следующие отображения, даже если он, возможно, будет знать алгоритм ГПСЧ и, может быть, текущее отображение. [46]

Безопасность данных мониторинга после вторжения, впрочем, не столь высока, как данных до вторжения. поскольку для первых надо обнаружить вторжение а также с уверенностью знать, что текущее двухбайтовое значение ГПСЧ было безвозвратно стерто. К счастью, впрочем, такое стирание может быть обычно выполнено за 2 мкс или меньше на скромном микропроцессоре. [47,48]

Обратите внимание, что стирание всего двух байтов ОБДПЗ происходит гораздо быстрее стирания ключей для КИП или обычных шрифтов с повышенной безопасностью длиной 128 или 256 байтов (и больше), соответственно. [49] Это может иметь важный подтекст для безопасности. [50] Более того, со стандартными «мусором», КИПом или шифром стирание ключа не гарантирует целостности данных, потому что теоретически «мусор», КИП или шифр могут быть раскрыты, особенно в случае с международными ядерными гарантиями, где противник, весьма вероятно, знает чистый текст, шифрованный текст и применяемый алгоритм, а также обладает огромными ресурсами. Напротив, полное стирание текущего двухбайтового значения ГПСЧ не оставляет противнику надежды надежной фальсификации будущей информации вне зависимости от усложненности его криптоаналитических способностей.

ТРЕБОВАНИЯ К ХРАНЕНИЮ ОБДПЗ

При 16 неповторяющихся гекс-цифрах в каждом отображении ОБДПЗ существует 16! = 2.1 x 1013 возможных отображений. Это соответствует минимуму в 44.3 бит, необходимому для представления любого возможного отображения.

Однако, с практической точки зрения наиболее простым и быстрым подходом станет хранение каждого отображения просто в виде 16 гексов, или 64 бит. Преимущество в том, что отображение превращается просто в таблицу для просмотра, не требующую ни вычислений, ни распаковки отображения микропроцессором в рабочих условиях. [51]

Слегка более эффективный путь для хранения отображений, который требует, впрочем, умеренных лишних вычислений микропроцессором, состоит в хранении каждого отображения в виде 15 гексов вместо 16. Последний гекс не нуждается в определении, так как он остался единственным, не появившемся пока еще в отображении. Такой подход требует только 15 гексов, или 60 бит на отображение.

Существует много других алгоритмов для представления отображений, которые требуют меньше памяти, но больше вычислений. Например, после того, как определены первые 8 гекс-цифр отображения, нужны еще три бита для определения следующей гекс-цифры из списка восьми (8=23) оставшихся и еще не выбранных гекс-цифр. После12 отобранных гекс-цифр нужны всего два бита, а после 14 отобранных – всего один бит. При таком подходе требуется 49 бит для полного определения отображения. Недостаток этого «алгоритма оставшихся цифр» в том, требуется больше счетного времени микропроцессора для распаковки каждого отображения, когда это потребуется.

Другой потенциальный алгоритм, называемый «высоким-низким алгоритмом», определяет каждую гекс-цифру в отображении максимально 4 битами. Эти биты указывают, находится ли данный гекс в нижней половине порядкового списка неотобранных цифр или в верхней. Половина гекс-цифр, не принадлежащая к заданной группе, удаляется и затем проверка повторяется. Эта процедура повторяется четыре раза, пока не все гекс-цифры не будут определены. По мере того, как отображение растет, порядковый список невыбранных гекс-цифр сокращается и поэтому на определение каждой гекс-цифры уже требуется меньше 4 бит. В среднем, этот алгоритм требует 46.4 бита на одно отображение, как указано в Табл. 1. [52]

Даже при самом неэффективном с точки зрения памяти алгоритме («Список 16 гекс-цифр»») ОБПДЗ не требует непрактичного объема памяти. Каждый гигабайт чистого текста будет нуждаться всего в16 гигабайтах отображений ОБПДЗ и эта память освобождается для других целей, как только завершается каждое отображение. Сейчас *розничная* цена памяти объемом в 16 гигабайт составляет менее3 долларов для памяти на твердом диске и менее 140 долларов для флэш-памяти. Эти цены не только меньше для устройств памяти при оптовой продаже, но стоимость памяти продолжает резко падать со временем, как показано на

Рис. 2. [53]

Табл. 1: Требования к памяти и распаковке для разных алгоритмов хранения отображений ОБДПЗ.

Сложность алгоритма	**Число бит**	**Число бит памяти на байт текста**	**Сложность распаковки**
Список 16 гекс-цифр	64	16	Никакой
Список 15 гекс-цифр	60	15	Минимальная
Остающиеся цифры	49	12.2	Умеренная
Высокий-низкий	46.4[1]	11.6[1]	Высокая
Теоретический минимум	44.3	11.1	Очень высокая
[1] В среднем			

Другой вопрос о памяти для микропроцессоров связан с количеством имеющейся памяти со случайным доступом (ПСД). Большинство современных шрифтов, которые можно попытаться установить на микропроцессоре, требуют от 50 до 2300 или даже больше байтов ПСД, а стандартные 8-битовые дешевые микропроцессоры обычно имеют ПСД в объеме 128 или 256 байтов. [28] Для схемы ОБДПЗ, показанной на Рис. 1, фактически потребуется всего 2 байта ПСД для текущего итеративного значения ГПСЧ.

Третий вопрос о памяти относится к требуемому объему для программирования. Многие современные схемы шифрования не годятся для текущих коммерческих микропроцессоров [28], а продвинутые алгоритмы КИП относительно велики. Напротив, при ОБДПЗ алгоритм шифрования – это просто таблица для просмотра. Ему нужны лишь несколько десятков строчек Бейзика для программирования и несколько сотен машинных инструкций – конкретный объем зависит от выбранной схемы распаковки отображений ОБДПЗ и сложности применяемого ГПСЧ.

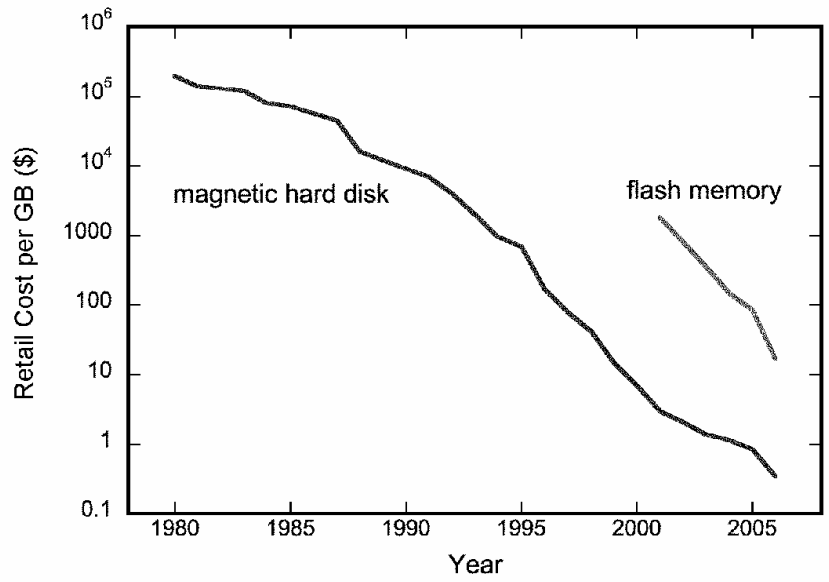

Рис.2: Указаны спадающие со временем (ось X) цены одного гигабайта памяти (ось Y) на твердых магнитных дисках (левый график) и на флэш-памяти (правый график). Если тенденция не изменится (даже без появления новых технологий хранения данных), окажется возможным хранить один гигабайт данных почти бесплатно к 2015 году.

СКОРОСТЬ ОБДПЗ

Шифрование с помощью ОБДПЗ осуществлялось на микропроцессоре Microchip PIC16F819 с двумя чипами памяти 24FC515 (512К бит). Математическое обеспечение было написано при помощи компилятора PICBASIC PRO. [54]

В такой системе счетное время, требуемое на шифровку одного байта чистого текста, составляло около 160 временных циклов независимо от скорости осциллятора – 4, 8 или 20 МГц. (Таким образом, на байт пришлось 8 мкс при 20 МГц). Более эффективный алгоритм и применение программирования на языке ассемблера вместо компилятора Бейзика, несомненно, увеличат скорость шифровки. Укажем для сравнения, что сильно оптимизированный шифровальный алгоритм Twofish требует 1820 временных циклов на байт для 8-битового микропроцессора, но предлагает относительно низкие уровни безопасности и ключ размером до 32 байта. [55]

СОЗДАНИЕ ОТОБРАЖЕНИЙ ОБДПЗ

Случайные отображения ОБДПЗ лучше всего можно генерировать с однократным блокнотом (ОБ) случайных гекс-цифр, которые в свою очередь создаются недетерминистично с использованием аппаратного обеспечения (обсуждаемого в следующем разделе). Создание ОБ на основе метода детерминистичного ГПСЧ (как найдено у большинства компьютеров) не приводит к чисто случайным числам и не рекомендуется, так как противник может обладать способностью предсказывать их поведение, особенно если большое количество отображений ОБДПЗ заложено на хранение в контролирующую аппаратуру. [56] Безопасность ГПСЧ окажется слабой, поскольку компьютерные ГПСЧ хорошо изучены, а кроме того существует только небольшое число других широко применяемых ГПСЧ [57]. Действительно, имеются известные примеры людей, предсказывающих значения ГПСЧ, то есть, «пробивающих» ГПСЧ. [58]

Проблема с применением ОБ со случайными гекс-цифрами для создания отображений ОБДПЗ состоит в том, что каждая цифра ОБ равновероятна, кроме того, каждое отображение ОБДПЗ должно быть составлено из 16 случайных гекс-цифр, каждая из которых появляется только однажды. Возможный алгоритм построения отображения ОБДПЗ из одного блокнота содержит простое добавление следующей случайной гекс-цифры из блокнота в отображение, если эта цифра еще не появлялась в отображении, и отбрасывание этой гекс-цифры с переходом к новой случайной гекс-цифре блокнота, если замечается повторение. Такой метод отличается прямолинейностью, образуя очень случайные отображения ОБДПЗ, если начальные гекс-цифры ОБ были случайными, но тратит в среднем 60.644 % +/- 0.023 % ОБ, чтобы избежать повторения цифр в отображении. Этот алгоритм называют «затратным алгоритмом». Пример приведен на Рис. 3.

Другой метод формирования отображения ОБДПЗ на основе ОБ со случайными гекс-цифрами называется «алгоритмом еще не использованного». Пример показан в Табл. 2.

Когда создается отображение ОБДПЗ, авторы следят, какие гекс-цифры еще не появились в отображении. Когда запускается новое отображение, этот «неиспользованный список» шифруется псевдослучайно с использованием компьютерного ГПСЧ. Каждая гекс-цифра в ОБ тогда используется, чтобы определить, какая гекс-цифра из «неиспользованного списка» присоединится к отображению. [59] Как только отображение завершено с 16 неповторяющимися гекс-цифрами, построение следующего отображения начинается снова с недавно зашифрованным «неиспользованным списком». Этот алгоритм не тратит ничего в ОБ, но он слегка менее безопасен в той степени, в какой все еще полагается на ГПСЧ компьютера. [60]

Случайные отображения ОБДПЗ могут также быть созданы, если позволить в значении ОБ перемешать случайным образом гекс-цифры (0123456789ABCDEF), использовав, например, алгоритм смешивания Фишера-Ятса. [61]

Другой подход назван «алгоритмом повторения отбора». Он использует аппаратурно генерированные случайные значения гекс-цифр ОБ, чтобы прерывисто повторять отбор внутреннего компьютерного ГПСЧ или других ГПСЧ. Значения, генерируемые ГПСЧ, используются затем для создания каждого случайного отображения. Этот алгоритм быстр и приводит

к высоко эффективному использованию ОБ. Однако он не максимально случаен, поскольку в его поведении доминирует ГПСЧ. Возможны также другие алгоритмы для создания отображений ОБДПЗ, помимо уже рассмотренных здесь.

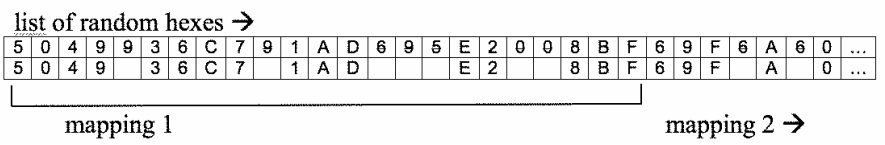

Рис. 3: Пример «затратного алгоритма для создания случайных отображений ОБДПЗ (нижний ряд) от аппаратурно генерируемого одноразового блокнота со случайными гекс-цифрами (верхний ряд). В любое время, когда одна из гекс-цифр из ОБ появляется в создаваемом отображении, она пропускается. В приведенном примере автору используют первую цифру 9 для отображения 1, но игнорируют вторую, третью и четвертую девятки, когда они появляются (поэтому повторяющиеся девятки вычеркиваются из отображения). Как только отображение полностью завершается шестнадцатью неповторяющимися гекс-цифрами, авторы создают следующее отображение. В этом примере завершенное отображение 1 имеет вид (504936C71ADE28BF), а отображение 2 начинается как (69FA0…..)

Независимо от алгоритма время вычислений для создания отображений ОБДПЗ может оказаться относительно умеренным у настольного компьютера. [62] Для компьютера Apple 2GHZ Intel Core Duo Mae с программами, написанными в REALbasic, [63] время создания двух миллионов отображений ОБДПЗ от существующего ОБ (достаточное для шифрования одного мегабайта чистого текста) составит 3.7, 1.1 и 3.1 минуты, соответственно, для «затратного алгоритма», «алгоритма еще не использованного» и смешивания Фишера-Ятса (16 смешиваний на отображение). Более быстрый компьютер в сочетании с более оптимизированными алгоритмами и компьютерными кодами значительно сократят эти времена.

ПОЛУЧЕНИЕ ИСТИННО СЛУЧАЙНЫХ ЧИСЕЛ

Непредсказуемость отображений ОБДПЗ критично зависит от непредсказуемости случайных чисел в ОБ, из которых отображения создаются. Вероятно, наибольшая случайность может быть получена на основе квантовых эффектов. [64] К счастью, истинно (или, по крайней мере, в высокой степени) случайные последовательности могут создаваться рядом различных способов с относительно недорогой аппаратурой.

Сюда входят измерения радиоактивного распада либо от фонового и излучения, либо от радиоактивного источника [65]., шумы электроники (тепловые шумы или шумы Джонсона, лавинные шумы, другие шумы в усилителях и квантовые электронные туннельные эффекты) [66], фотоны, проходящие через расщепитель (50 на 50) пучка [67], шумы в закрытой видеокамере, микрофонные шумы [68], свет от люменисцентных ламп [69], радиочастотные шумы (из ненастроенного приемника) [69], шумы твердого диска компьютера [70], шарики из лотерейных машин [71] и непредсказуемость выбора момента, когда человек пользуется клавиатурой или мышкой компьютера [72]. В Табл. 3 подведены итоги некоторых из этих методов.

Пять из восьми методов, приведенных в Табл. 3, используются автором для создания случайных чисел, но самым легким оказалось использование фотодетекторов для слежения за дисками и сферами плазменного разряда, которые продаются как лампы-новинки (смотрите Рис. 4). До четырех разных фотодетекторов можно применять для создания отдельных некоррелированных и случайных последовательностей на лампу. Эта новая методика безопасна, проста и недорога, а также может генерировать случайные числа с умеренно высокой скоростью. Она также совсем нечувствительна к проблемам калибровки и дрейфа порога, а также к вскрытию на расстоянии. Созданные таким образом цифровые последовательности согласованно прошли все испытания на случайность, проведенные с ними.

Табл. 2: Пример алгоритма «Еще не использованного» для создания случайного отображения ОБДПЗ от одноразового блокнота случайных гекс-цифр.

Шаг	А	Б	В
1	C	EB46C1572A0D39F8	C
2	1	EB461572A0D39F8	CB
3	0	E461572A0D39F8	CBE
4	F	461572A0D39F8	CBE1
5	0	46572A0D39F8	CBE14
6	A	6572A0D39F8	CBE148
7	4	6572A0D39F	CBE148A
8	2	65720D39F	CBE148A7
9	8	6520D39F	CDE148A70
10	1	652D39F	CDE148A705
11	6	62D39F	CDE148A7056
12	0	2D39F	CDE148A70562
13	7	D39F	CDE148A70562F
14	3	D39	CDE148A70562FD
15	5	39	CDE148A70562FD9
16	-	3	CDE148A70562FD93

На первом этапе авторы только что выбрали первую гекс-цифру для нового отображения, равную доступной случайной цифре из ОБ (в данном случае. С). На этапе 2 следующая случайная гекс-цифра из ОБ равна 1. Она используется для указания очередной гекс-цифры в положении 1 (псевдослучайно обрезанного) «неиспользованного списка», которая в данном примере равна В, так что В присоединяется к отображению. (Первая цифра в «неиспользованном списке» определяется как положение 0). На этапе 3 следующей цифрой «неиспользованного списка» в положении 0 оказывается Е. На этапе 4 положение F соответствует положению 15, но в «неиспользованном списке» осталось всего 13 гекс-цифр, так что авторы свертывают положение 15 в положение 2, что соответствует гекс-цифре 1. Процесс продолжается до этапа 16, когда единственная оставшаяся гекс-цифра в «неиспользованном списке» (в этом случае 3) присоединяется к отображению. Для создания нового отображения авторы создают новый случайный обрезаемый «неиспользованный список» и получают новые случайные гекс-цифры от ОБ.

А - Случайный гекс из ОБ

Б - Обрезаемый «неиспользованный список»

В - Отображение в процессе создания

УСТАНОВКА ОТОБРАЖЕНИЙ ОБДПЗ

Отображения ОБДПЗ должны быть установлены или отправлены на контролирующую аппаратуру инспекторами таким образом, чтобы не подвергать риску безопасность. Один метод заключается в том, чтобы сохранить случайные отображения на «флэшке», которая скопируется в контролирующую аппаратуру в рабочих условиях лично инспекторами, когда они в первый раз начнут свои дела. «Флэшка» затем будет полностью стерта или оставлена на сохранение инспекторам. Если хозяин (инспектируемая страна) встревожится по поводу того, какая информация может поступать в аппаратуру, можно применить схему «выбора или сохранения» [73], где инспекторы выкладывают три или пять «флэшек» (каждая со своими случайными отображениями ОБДПЗ) и инспектируемая сторона наугад указывает, какую следует установить, а затем выбирает еще одну, чтобы поменять настройку. Последняя не будет использована для мониторинга, но может быть вместо этого проверена хозяином, чтобы убедиться в наличии на ней только случайных отображений, но не кода микропроцессора или неслучайной информации.

Недорогая «флэшка» на 4 ГБ может хранить до 34 миллионов отображений ОБДПЗ, чего хватает для зашифровки 182 МБ данных чистого текста. [74] Если одно (0-65535) измерение производится и записывается в течение каждой секунды, то достаточно отображений почти

для трех лет мониторинга!

Табл. 3: Характерные черты различных способов физического создания случайных чисел аппаратурой. За исключением люминесцентных ламп и отдельных шумов электронной методики все остальные по своей природе целиком или частично являются квантовомеханическими[3].

Метод	**Типичная скорость генерации (бит/мин)**	**Типичная стоимость**
Клавиатура и мышка	2 - 100	$0
Люминесцентная лампа	10^1 - 10^6	$100-$600
Фоновый радиоактивный распад	5 - 10	$300
Радиоактивный распад источника	20 – 6,000	$350-$800
Плазменный диск или сфера	200 - 1000	$100
Радиошумы	10^3 – 10^4	$500
Фотоны и расщепитель пучка	10^6 - 10^8	$2000
Электронные шумы	$2x10^4$ – $2x10^8$	$300-$3500

[1] Для значений, приведенных в этой строке, использованы 4 независимых фотодиода для контроля за световым выходом одного диска или лампы с плазменным разрядом.
[2] Типичные расходы на аппаратуру и датчики включают их розничную цену, но не учитывают компьютер или устройство для сопряжения, использованные для получения и записи случайных цифр.
[3] Плазменный разряд по сути относится к квантовым явлениям, но конкретное пространственное и временное поведение плазменного разряда зависит кроме того от космических лучей, которые имеют заметный квантово-механический характер. Является ли методика клавиатуры и мышки полностью или частично квантово-механической, зависит от степени, до которой человеческие существа и их решения определяются квантово-механическими событиями.

ЗАКЛЮЧЕНИЕ

Необходимо улучшить обеспечение безопасности информации по ядерному мониторингу с учетом текущей ненадежности пломб с индикацией взлома и не столь высокой безопасности, предлагаемой обычными методами идентификации данных, будь то «мусор», КИПы или шифры. ОБДПЗ представляется привлекательной альтернативой. В отличие от обычных методов целостности данных зашифрованная при помощи ОБДПЗ информация, которая записывается (или передается) до вторжения, полностью безопасна (не только с точки зрения вычислений), даже если аппаратура мониторинга не смогла обнаружить вторжение и ничего не стерлось. Данные, поступившие после вторжения, безопасны, если вторжение может быть обнаружено и быстро стираются всего лишь 2 байта. Другие методики целостности данных, напротив, требуют стирания, по крайней мере, 128 байтов, а иногда значительно больше.

ОБДРЗ имеет и другие преимущества. Он очень быстр и прост с вычислительной точки зрения – в основном, стираемая таблица для просмотра. Будучи таким прямолинейным и несложным, ОБДПЗ открыт для прозрачности, простоты и высоких уровней комфорта, что так важно для международных ядерных гарантий. Его простота должна также сделать ОБДПЗ относительно невосприимчивым к атакам противника по «боковым каналам», например, выбор определенного времени или методы анализа мощности [75], а также к выбранным атакам на чистый текст. [17] ОБДПЗ не связан с большими количествами ПСД, объема кодов или времени вычислений в микропроцессоре и вполне практичен для осуществления на дешевых 8-битовых микропроцессорах. В отличие от ряда современных КИПов и шифров ОБДПЗ не имеет патентных, лицензионных и экспортных проблем. Более того, его безопасность не подвергается опасности (в отличие от некоторых других методов шифрования и целостности данных), когда один тот же чистый текст послания шифруется более одного раза.

Относительно большое количество информационной памяти, требуемой для ОБДПЗ, фактически предлагает преимущества в безопасности. Если бы противник использовал усложненные методики для попыток восстановления стертых отображений ОБДПЗ (это значительное опасение [49]), чтобы он мог изменить данные мониторинга, полученные после вторжения, ему пришлось бы попытаться восстановить очень большой объем информации, что делает его задачу более сложной.

Рис. 4 : Потребительский **д**вумерный диск (слева) и сферическая трехмерная лампа на Будде (справа). Лни продаются в розницу как игрушки и украшения за 19 и 25 долларов, соответственно. Нити плазменного разряда, которые генерирует каждое устройство, флуктуируют непредсказуемо в пространстве и времени. До 4 разных фотодатчиков при правильном расположении могут измерять уровень свечения различных точек каждого устройства без любой значительной корреляции или антикорреляции в их измерениях.

Среди недостатков ОБДПЗ требования иметь11-16 байтов данных отображения на байт чистого текста, подлежащего шифровке (хотя этот объем памяти освобождается для других использований по мере создания шифрованного текста). ОБДПЗ также требует большого количества недетерминистских, аппаратурно-созданных случайных чисел и установки информации по отображениям ОБДПЗ в аппаратуре мониторинга таким образом, который сохраняет ее в секрете. В отличие от шифров с публичными или частными ключами ОБДПЗ не способствует идентификации третьей стороной. [76]

ОБДПЗ был осуществлен как на настольном компьютере, так и на недорогом 8-битовом

микропроцессоре. Авторы нашли, что его легко разрабатывать и применять.

ПРИМЕЧАНИЯ И ССЫЛКИ

1. Roger G. Johnston and Morten Bremer Maerli, “International vs. Domestic Nuclear Safeguards: The Need for Clarity in the Debate Over Effectiveness,” *Disarmament Diplomacy* 69 (2003): 1-6. http//www.acronym.org.uk/dd/dd69/ 69op01.htm; Morten Bremer Maerli and Roger G. Johnston, “Safeguarding This and Verifying That: Fuzzy Concepts, Confusing Terminology, and Their Detrimental Effects on Nuclear Husbandry,” *Nonproliferation Review* 9 (2002): 54-82, cns.mils.edu/pubs/npr/vol09/91/91maerli.pdf
2. Roger G. Johnston,”Tamper-Indicating Seals,” *American Scientist* 94 (2006): 515-523.
3. Роджер Дж.Джонстон, «Пломбы с индикацией вмешательства для ядерного разоружения и обращения с опасными отходами», *Наука и всеобщая безопасность* 9 (2) (2001): 9-19.
4. Daniel Engber, ”How Do You Seal a Nuclear Plant?” http://www.slate.com/id/2123769/
5. При рассмотрении безопасности контролируемых данных, передаваемых в реальном времени, привлекают внимание два фактора: время задержки при поступлении данных в принимающий центр (чем потенциально может воспользоваться противник) и тот факт, что данные в реальном времени часто записываются и хранятся (по крайней мере, временно) до того, как анализируются.
6. Jon S. Warner and Roger G. Johnston, “GPS Spoofing Countermeasures,” Homeland Security Journal, December 12, 2003, http://www.homelandsecurity.org/bulletin/Dual%20 Benefit/warner,gps.spoofing.html
7. Roger G. Johnston, ”Tamper Detection for Safeguards and Treaty Monitoring:: Fantasies, Realities, and Potentials,” *Nonproliferation Review* 8 (2001): 102-115, http://www.princeton.edu/globsec/publications/pdf/9.2johnston.pdf
8. Roger G. Johnston and Anthony R. F. Garcia, “An Annotated Taxonomy of Tag and Seal Vulnerabilities,” *Journal of Nuclear Materials Management* 229 (2000): 23-30.
9. Roger G. Johnston, Anthony R. F. Garcia, and Adam N. Pacheco, “Efficacy of Tamper-Indicating Devices,” *Journal of Homeland Security,* April 16, 2002, htpp://www. homelansecurity.org/journal/Articles/displayarticle.asp?article=50
10. Roger G. Johnston, “The ‘Anti-Evidence’ Approach to Tamper-Detection,” *Packaging, Transport, Storage* & *Security of Radioactive Material* 16 (2005): 135-143.
11. Оценка анти-информации в данной пломбе (или на ней) засекречена и нет уверенности, что она использована. Обычно анти-информация меняется на новую непредсказуемую оценку, когда пломба используется заново.
12. Методика АИП кроме применения для пломб обладает рядом важных преимуществ для надзора и мониторинга в реальном времени. Этот метод называется “Town Crier” (“Городской глашатый”) – смотрите Roger G. Johnston, Anthony R. F. Garcia, and Adam N. Pacheco, “The “Town Crier” Approach to *Monitoring*,” *International Journal of Radioactive Material Transport* 13 (2002):117-126; Roger G. Johnston, Anthony R. F. Garcia, and Adam N. Pacheco, “Improved Security Via “Town Crier” Monitoring, ” Proceedings of Waste Management’03, Tucson, AZ, February 24-27, 2003, waww.osti.gov/servlets/purl/827636-nWiBFO/native/. При использовании вместе с ОПДЗЗ метод «городского глашатая» требует не такого надежного детектирования физического или электронного вскрытия, чем при обычных подходах при мониторинге в реальном времени.
13. Если исходить из опыта автора, то обычное использование механических замков для шкафов с оборудованием, чтобы зафиксировать вторжение, все равно, чтобы вообще не фиксировать вторжение.
14. Авторы благодарны анонимному рецензенту за привлечение внимания к важности четкого разграничения между целостностью информации и конфиденциальностью информации. Первое является основной темой данной статьи, даже хотя авторы (необычно) предложили применение шифра для гарантии целостности информации.
15. Alfred J. Menezes, Paul C. van Oorshot, and Scott A. Vanstone, *Handbook of Applied Cryptography* (New York: CRC Press, 1996).
16. Douglas R. Stinson, *Cryptography: Theory and Practice* (Boca Raton, FL: CRC Press, 1995).
17. Bruce Schneier, *Applied Cryptography* (New York: Wiley, 1995).

18. Иногда говорят, что «шифровка не обеспечивает целостности данных» (смотрите, например, Brad Conte, “Hashes”, http://b-con.us/security/hashes.php). Это высказывание справедливо, но только в педантичном, формальном смысле. Если противник собирается делать случайные изменения в зашифрованном тексте без понимания влияния на расшифрованный текст (а в начальном тексте не предусмотрены проверка суммы знаков или иные хитрости), то при расшифровке данных нельзя будет быстро определить такие случайные вскрытия зашифрованного текста. Но при ядерном мониторинге задачей нечестного противника станет замена истинной информации мониторинга, указывающей на обман, на фальшивую информацию, указывающую на номинальные условия. Например, если оператор установки хочет пронести радиоактивные материалы мимо радиологического монитора на выходе, он может захотеть заменить уровни высокой радиации, регистрируемые контролирующей аппаратурой, на нормальные фоновые отсчеты. Случайный вандализм с зашифрованной информацией не позволит оператору выполнить такие действия. Более того, информация, занесенная в исходный текст вместе с ОПДЗЗ, будет почти наверняка содержать вместе с фактическими контрольными измерениями метки времени, номера измерений, контрольные суммы цифр, четность или прочие простые хитрости, которые легко позволят обнаружить случайные изменения или перестройку данных.
19. Зашифровка или идентификация данных также могут использоваться для связи между двумя разными людьми (даже для связи с самим собой), разделенными во времени, но находящимися в том же самом физически безопасном месте. Это типичный сценарий для безопасности архивов компьютерных данных.
20. Типичное заявление относительно безопасности «мусора» или кодов идентификации посланий (КИП) таково – « до сих пор мы не слышали, чтобы кто-нибудь справился с ними». Это не очень хорошее мера безопасности, особенно потому, что власти, пробившие «мусор» или КИП, вряд ли будут публично этим хвастать, ибо в ином случае будет ограничена их возможность использования своего открытия. Напротив, для большинства современных шифров у математиков существует точное представление о сроках и компьютерной мощности, требуемых для разгадки шифра (по крайней мере, с использованием обычных методов). Конечно, проблема не в знании того, когда или если произойдет крупный криптографический прорыв. С одноразовым блокнотом (или ОПДЗЗ) не стоит бояться в будущем новых криптоаналитических методов из-за математически доказанной их непробиваемости.
21. Protechnix, “Cryptology and Data Security: The Vernam Cipher,“ http://www.prothenix.com/information/crypto/pages /vernam.base.html
22. Смотрите, например, David A. McGrew and Scott R. Fluhrer, “Multiple Forgery Attacks Against Message Authentication Codes,” http//eprint.iacr.org/2005/161.pdf; Bruce Schneier, “Schneier on S:ecurity,” http://www. Schneier.com/blog/archives/2005/02/sha1.droken.html; Bruce Schneier, ”*Secrets and Lies: Digital Security in a Networked World. (*New York: Wiley, 2000); Stefan Wolf, “Unconditional Swcurity in Cryptography,” in *Lectures on Data Security: Modern Cryptology in Theory and Practice,* ed. Ivan Damgard (New York: Springer Ferlag, 1999):217-250; Simon Singh, *The Code: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography* (New York: Doubleday,1999); Fred B. Wrixon, *Codes, Ciphers, & Other Cryptic & Clandestine Communication* (New York: Black Dog & Leventhal, 1998).
23. Sarah Granger, “Unlocking the Secrets of Crypto,” http:/www.securityfocus.com/infocus/1617
24. Хотя микропроцессоры могут обычно стереть единичный байт за 2 мкс или скорее, на стирание многих байтов могут потребоваться многие миллисекунды. Взлом электронных схем вполне возможен менее, чем за мс, если установить посторонние контакты с электроникой, отсоединить силовое питание или даже выстрелить через определенную точку, чтобы разрушить контакты или бит безопасности. Более того, микропроцессоры обычно испускают заметные электромагнитные сигналы, которые трудно полностью заблокировать. Это излучение можно использовать для определения того, чем занят микропроцессор в данный момент, и это позволяет противнику точно выбрать время для нападения даже на уровне долей микросекунды.
25. Обычно для противника будет гораздо труднее узнать точный чистый текст контрольных измерений, которые обладают шумом или большим динамическим диапазоном, но это не относится к простым измерениям, например, обнаруживает ли датчик (или не обнаруживает) сигнал, превышающий порог тревоги

26. Даже если противнику международных ядерных гарантий неизвестен использованный алгоритм шифрования или идентификации данных, в также присвоенный микропроцессору код, он сможет узнать в любом случае используемый алгоритм, потому что согласно правилу.Шеннона противник обычно может его вычислить. Смотрите Roger G. Johnston, “Cryptography as a Model for Physical Security, ”*Journal of Security Administration* 24 (2001):33-43; and Claude E. Shannon, “Communication Theory of Secrecy Systems, “*Bell System Technical Journal* 28 (1949): 656-715.
27. Gustavus J. Simmons, *Contemporary Cryptology: The Science of Information Integrity* (New York: Wiley-IEEE Press, 1999), 73ff; Bruce Schneier, *Applied Cryptography* (New York: Wiley, 1995), 17-18.
28. Bruce Schneier, John Kelsey, Doug Whiting, et al., “Performance Comparison of the AEC Submissions,” Proceedings of the Second AES Candidate Conference, NIST, March 1999, 15-34, http://www.schneier.com/paper-aes-performance.html
29. Строгие бюджетные ограничения (см. [7]), наложенные на МАГАТЭ, увеличили важность применения дешевых компонентов для аппаратного обеспечения мониторинга.
30. Смотрите, например, Stefan Wolf, “Unconditional Security in Cryptography,“ in *Lectures on Data Security: /Modern Cryptology in Theory and Practice,* ed. Ivan Damgard (New York: Springer-Verlag, 1999), 217-250; and “One Time Pad” in *Glossary of Cryptographic Terms* (2000), http://www..pgp.net/pgpnet/pgp-faq/pgp-faq-glossary.html.
31. Шифр Вернана называется одноразовым блокнотом (ОБ), потому что случайные цифры кода вначале записывались на блокноте. Когда все цифры на данной странице использовались, она отрывалась от блокнота и уничтожалась.
32. David Kahn, *The Code-Breakers* (New York: Scribner, 1996). Без ОБ захваченные зашифрованные документы никогда не были дешифрованы (и не будут).
33. «Модули» или «функции свертки» - это остаток от деления на целое число. Так, например, 9mod10=9, 10mod10=0, 11mod10=1. Аналогично, 99mod100=99, 100mod100=0, 101mod99=1 и так далее.
34. ОБ может быть использован как часть машинного кода идентификации (МКА) или же для шифровки «мусора», Смотрите, например, Douglas R, Stinson, C*ryptography: Theory and Practice* (Boca Raton, FL: CRC Press, 1995). Однако это не снимает проблему быстрого стирания многих байтов при обнаружении несанкционированного доступа к оборудованию для мониторинга. Более того, применение зашифрованного «мусора» или МКА, основанных на ОБ, не гарантирует того, что противник не может раскрыть фальшивые данные, находящиеся в незашифрованном тексте и говорящие о нарушении Договора, что приведет к неизменившимся значениям ярлыка МКА или «мусора». Последнее тоже оказывается проблемой, если ОБДЗЗ использовать в МКА или для шифровки «мусора».
35. Fred B. Wrixon, *Codes, Ciphers & Other Cryptic & Clandestine Communication* (New York: Black Dog & Leventhal, 1998):168-237.
36. Таким образом, две шестерки в «1663» чистого текста обе зашифрованы в 0 в зашифрованном тексте «9003».
37. Шифры подстановки, используемые для зашифровки слов, легко могут быть раскрыты с использованием карандаша и бумаги (плюс еще некоторое знание используемого языка). Обычно хватает одного-двух предложений зашифрованного текста.
38. Как и шифр Вернана, ОБДЗЗ представляет из себя поточный шифр, где цифры (или символы) начального текста трансформируются независимо друг от друга. Но в отличие от шифра Вернана каждая цифра в зашифрованном текста связана с соответствующей цифрой чистого текста нетривиальным неаддитивным способом.
39. В этом примере, если противник захотел записать ложные данные мониторинга «0000» вместо истинного значения «1663» в исходный текст (возможно, это нулевое значение радиологического сигнала вместо увеличенного показания), прошлые стирания отображений ОБДЗЗ по мере их использований поставит его перед неразрешимой дилеммой. Он знает, что первая «1» чистого текста преобразуется в цифру «2» зашифрованного текста при первом отображении, потому что он имеет доступ к полному зашифрованному тексту после проникновения в контрольную аппаратуру, но остаток первого отображения уже исчез. Нет информации, чтобы помочь ему в решении, как должна преобразоваться цифра «0» чистого текста. Та же проблема относится к остальным трем цифрам чистого текста. И если даже у него существуют правильные догадки относительно первого отображения, это не поможет

ему догадаться об остальных трех отображениях. Применение разных типов контрольных сумм, четности, простых лишних данных, указания времени или номера измерения, которые все будут зашифрованы с помощью ОБДЗЗ (вместе с фактическими данными мониторинга), еще более усложнят попытки противника фальсифицировать прошлую или будущую информацию.

40. Шансы угадать правильную десятичную цифру в зашифрованном тексте, соответствующую данной десятичной цифре чистого текста, равны 1:9, а не 1:10. Дело в том, что противник уже знает одну из цифр каждого отображения, если предположить, что ему известны оба (чистый и зашифрованный) текста в результате тайного проникновения в контролирующую аппаратуру, (Конечно, если он знает только зашифрованный текст, а не чистый, шансы составят 1:10). При шансе 1:9 для догадки о верной цифре зашифрованного текста шанс правильной догадки пяти последовательных цифр равен 1:59000. Применение гекс- цифр (что более практично для микропроцессоров) вместо десятичных изменяет шансы для одной цифры до 1:15 и 1:760000 для пяти последовательных цифр.

41. Гексы (основание 16) – это наиболее практичный и естественный выбор для микропроцессоров независимо от размера слова микропроцессора, поскольку компьютеры и микропроцессоры сейчас являются двоичными устройствами и останутся такими на обозримое будущее. Каждая цифра гекса может быть представлена четырьмя битами и принадлежит к набору 0.1,2,3,4,5,6,7,8,9,A,B,C,D,E,F:. Например, гекс A = двоичное число 1010 = десятичное 10, гекс F = двоичное 1111 = десятичное 15. Простой байт (состоящий из восьми битов) может быть представлен двумя гекс-цифрами. Поэтому десятичное 256 = гекс FF. Если вместо применения гекс-чисел в качестве фундаментальных единиц будут использоваться для ОБДЗЗ байты, каждое отображение окажется на 256 байтов длиннее, а это потребует значительно больше памяти.

42. Полезно обезопасить данные, полученные до посягательства противника, даже если полученные после этого данные могут оказаться подвергнутыми риску, потому что они ограничиваются при возможном нападении. Обычно инспекторы присутствуют при начале мониторинга и в течение некоторого время после этого, чтобы проверить правильность работы аппаратуры.

43. Смотрите, например, Janes E. Gentle, *Random Number Generation and Monte Carlo Methods* (New York: Springer, 2003): 1135; и William H. Press, Saul A. Teukolsky, William T. Vettering, and Brian P, Flannery, *The Art of Scientific Programming* (Cambridge University Press, 1982), 274ff. Для приложений ОБДЗЗ этот «ключ» и также все последующие вычисленные псевдослучайные числа будут, по-видимому, обладать длиной в два байта.

44. Обратите внимание, что линейные конгруентные генераторы (ЛКГ) обычно считаются очень плохим выбором для ГПСЧ и криптографической безопасности, так как часто оказывается относительно легким предсказать будущие результаты после изучения нескольких их предыдущих результатов. В данном случае это не вызывает опасений из-за присущего им вырождения (избыточности) [46] и поскольку противник имеет доступ только к одной или двум точкам информации (в зависимости от того, где он взаимодействовал с циклом микропроцессора). Нельзя предсказать результаты ЛКГ с таким малым числом информационных точек даже без врожденного вырождения.

45. Вместо использования ОБДЗЗ можно применять концепцию автора относительно вырожденного ГПСЧ, который можно использовать для указаний различным ключам КИП, которые подлежат стиранию после каждого применения. Это потребует меньше памяти по сравнению ОБДЗЗ, но (в отличие от ОБДЗЗ) больше машинного счета и не гарантирует достоверности данных, полученных до вмешательства, так как всегда остается вероятность (не имеет значения, насколько она мала) того, что противник сможет генерировать фальшивые данные на основе того же самого флага КИП.

46. Хотя текущее значение ГПСЧ подлежит стиранию. как только обнаружится постороннее вмешательство, но противнику известен алгоритм ГПСЧ и он попытается заморозить микропроцессор в середине процесса обмена или стирания, так что он способен, в принципе, перестроить расчеты ГПСЧ. Эта уязвимость может быть в значительной степени уменьшена из-за «вырождения», когда выход ГПСЧ получает более высокую точность, чем число, требуемое для выбора следующего отображения. Например, двухбайтовые (0-65535) значения, как предлагается в этой статье, могут быть объединены с вычислениями по модулю 100 для определения того, какое отображение (0-99) из небольшой кэш-памяти на Рис.1 будет ото-

брано следующим. Таким образом, в среднем имеется 655 различных значений ГПСЧ, которые приводят к изменению такого же числа отображений. Такая избыточность означает, что противник не сможет надежно переделать намеченные следующие значения ГПСЧ, просто зная текущее отображение, или то, которое будет выбрано следующим. Авторы успешно продемонстрировали методику избыточности в применении к АИП типа временной ловушки: Roger G. Johnston, 'Anti-Evidence Seals," Los Alamos laboratory Report LAUR-06-1312 (February 2006). В этом приложении имеются в среднем 400 различных значений ключа, которые дают один и тот же результат по «мусору» в данное время. Поэтому противник, знающий алгоритм «мусора» (но не зная его ключа), имеет шанс правильной догадки о «мусоре» на будущие времена, равный только 1:400 (0.25%).

47. Противник, озабоченный тем, что зашифрованные данные мониторинга демонстрируют его обман или неумение скрыть попытку взлома, может всегда, конечно, стереть зашифрованный текст или утерять, повредить или разрушить элементы механического обеспечения мониторинга, но результаты таких действий будут замечены инспекторами.

48. Обратите внимание, что стирание алгоритма ГПСЧ само по себе не оказывается необходимым по мере того, как стерты результаты его текущих вычислений. Способность рассказать инспектируемой стороне о реальном алгоритме ГПЧС (или даже показать его) может иметь важные последствия в отношении прозрачности международных ядерных гарантий.

49. Гарантирование полного стирания информации из памяти или из хранения не является тривиальной проблемой. Смотрите, например, Peter Gutmann, "Data Remanence in Semiconductor Devices, http://www.cypher punks.to/~peter/usenix01.pdf; Peter Gutmann, Proceedings of the Sixth USENIX Security Symposium, San Jose, CA, July 22-25, 1996. http://www.cs.aukland.ac.nz/%7Epgut001 /pubs/secure.del,html.

50. Чем быстрее идет стиранее, тем менее вероятно, что противник сможет прервать его, и тем больше шансов имеет микропроцессор, чтобы переписать память много раз для удаления остатков данных, и тем лучше безопасность. Предпочтительнее стирать быстрее максимальной механической скорости: звуку требуется около 15 мкс, чтобы пройти микрочип длиной 5 мм. Значение двухбайтового слова ГПЧС для ОБДЗЗ может быть стерто из памяти микропроцессора за 2 мкс или быстрее. Напротив, стирание ключа размером 256 байт (который часто не будет совпадать с памятью микропроцессора), изготовленного на фирме EEPROM, обычно занимает свыше 120 мкс.

51. Отображения не могут при хорошей безопасности создаваться детерминистическим ГПСЧ, встроеннымв аппаратное обеспечение мониторинга. Даже если текущие значения ГПСЧ могут быть удалены немедленно после обнаружения вторжения (иногда этого нельзя гарантировать), у противника будет достаточно информации от алгоритма ГПСЧ, обширного чистого текста и записанного зашифрованного текста, чтобы вычислить прошлые и будущие отображения. Если алгоритм ГПСЧ может оставаться секретным и надежно удаляемым (что иногда несовместимо с прозрачностью национальной безопасности хозяина), утонченный противник будет еще способен вычислить отображения, особенно если имеются большие количества зашифрованного текста. Конечно, нельзя создавать отображения в нужном виде недетерминистической аппаратурой ГПЧС, так как отображения тогда будут неизвестны инспекторам.

52. Одно ОБДЗЗ-отображение требуется для каждой гекс-цифры, подлежащей зашифровке, и два отображения - для каждого байта исходного текста. Сложные алгоритмы сжатия (например, зип-сжатие) могут еще больше уменьшить объем памяти, требуемый для ОБДЗЗ-отображений, обычно на 5-10 %, но сильно увеличивают расчетную сложность распаковки.

53. Частично заимствовано из "Historical Notes about the Cost of Hard Disk Storage Space," file:///Volumes/Corsair %202GB/Cost%20of%20Drive%20Space.webarchive and Thomas M. Coughlin< "Flash Illuminates Mobile Digital Storage," http://www.entertainmentstorage.org/articles/A%20 storage%20vision%202.pdf

54. microEngineering Labs, http://www.melabs.com

55. Bruce Schneier, "Twofish: A 128-Bit Block Cipher," http;//www.schneier.com/paper-twofish-paper.html

56. Великий математик Джон фон Нейман (1903-1957) однажды заметил, что «Любой, кто пытается создать случайные числа детерминистическими средствами, конечно, живет в состоянии греха». Взято из Herman H. Goldstine, *The Computer from Pascal to von Neumann* (Princeton, NJ: Princeton University Press, 1972), 378

57. William H. Press, Saul A. Teukolsky, William T, Vettering, and Brian P. Flannery, *Numerical Recipes in* C: *The Art of Scientific Programming* (Cambridge University Press, 1982), 274ff.
58. Kevin D. Mitnick and William L. Simon, *The Art of Intrusion: The Real Stories Behind the Exploit of Hackers, Intruders, & Deceivers* (New York: Wiley, 2005); J. Kelsey, B. Schneier, D. Wagner, and C. Hall, " Cryptoanalytic Attacks on Pseudorandom Number Generators," Fast Software Encryption, Fifth International Workshop Proceedings, March 1998, Springer-Verlag, 168-188, htpp://www.schneier.com/paper-prings.pdf; Peter Gutmann, "Software Generation of Practically Strong Random Numbers," https://www.usenix.org/publications/library/ proceedings/sec98/summaries/
59. Шестнадцатый шаг не требуется, если используется «Алгоритм списка пятнадцати гекс-цифр» для хранения отображений в микропроцессоре, который применяется для полевого мониторинга. Микропроцессор может определить последнюю гекс-цифру в каждом отображении, поскольку всего одна из 16 гекс-цифр еще не появилась в отображении.
60. Поскольку в конечном итоге каждое отображение зависит от случайного списка гекс-цифр, созданного недетерминистским образом аппаратурным обеспечением (а не ГПСЧ), тот факт, что «неиспользованный список» шифруется псевдослучайно компьютером, не должен сильно подвергать риску безопасность.
61. Paul E. Black, "Fisher-Yates shuffle," in Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed.. U.S. National Institute of Standards and Technology. http://www.nist.gov/dads/fisherYatesShuffle.html, December 19, 2005. Обратите снимание, что 16 перестановок, например, потребуют 32 цифры из ОБ, так что в данном случае ОБ используется не очень эффективно.
62. Создание отображений при помощи ОБДЗЗ, конечно, производится руководством до запуска аппаратуры для мониторинга, так что время работы компьютера не столь критично по сравнению с тем, когда приходится это делать в реальном времени в рабочих условиях на микропроцессоре.
63. REALbasic – это язык компилятора и программы, разработанные фирмой Real Software, http://realsoftware.com.
64. Полнота квантовой механики, как кажется, предполагает, что даже вся Вселенная не сможет предсказать случайные числа, создаваемые квантовыми эффектами.
65. Смотрите, например, John Walker, "HotBits: Genuine Random Numbers Generated by Radioactive Decay," http://www.fourmilab.ch/hotbits/; Black Cat Systems, "Generating Random Numbers Using Radiation," http://www.blackcatsystems.com/GM/rabdom.html; Wired Online, "Totally Random," http://www/wired.com/wired/ archive11.08/random.pr.html
66. Для примеров коммерческих электронных ГПСЧ смотрите Orion Products, "Random Number Generator," http://www.randomnumbergenerator.nl; ComScire. "Design Principles and Testing of the QNC Mode J1000KU," http://comscire.com/Products/J1000KU/; "Hardware Random Bit Generator," http://willware.net:8080/hw-rng.html; and Protego, "SG100 TRNG," http://www.protego.se /sg100_en.htm
67. Смотрите, например, Quantique, "Quantum Random Number Generators," http://www.idquantique.com/products/quantis.htm; Thomas Jennewein, Ulrich Achleitner, Gregor Weihs, Harald Weinfurter, and Anto Zellinger, "A Fast and Compact Quantum Random Number Generator," R*eview of Scientific Instruments* 71 (2000): 1675-1680; Ma Hai-Quiang, Wang Su-Mei,.Chang Jun-Tao, Ji Ling-Ling, Hou Yan-Hue, and Wu Ling-An, " A Random Number Generator Based on Quantum Entangled Photon Pairs," *Chinese Physics Letters* 21 (2004): 1961-1964, http://www.ingentaconnect/iop/cpl/2004/00000/ 21/00000010/art00027.
68. Mahesh Joharl, "Really Random Numbers," http://www.dirfile.com/really_random_numbers.htm
69. Wired Online, " Totally Random, "http//wired.com/wired /archive11.08/random.pr.html
70. Don Davis, Ross Ihaka, and Philip Fenstermacher, "Cryptographic Randomness from Air Turbulence in Disk Drives," http://world,std.com/~dtd/random/forward.pdf.
71. Richard P. Dunnigan, "Random Number Generator," U.S. Patent 4,786,056.
72. Истинно случайная последовательность не может быть создана на основе англоязычного текста, Смотрите, например, Claude E. Shannon. "Prediction and Entropy of Printed English, ' *Bell System Technical Journal* 30 (1951): 50-64, из-за структур, присущих любому языку. Создание случайных или псевдо-случайных предложений при использовании человеком компьютерной клавиатуры или мышки основано на тех микросекундах, когда нажимают клавишу или щелкают мышкой. Авторы применяли аналогичную методику во многих своих пломбах,

фиксирующих вторжение. Смотрите Roger G. Johnston, “Anti-Evidence Seals,” LANL Report LAUR-06-1312, February 2006. Действия человека определяют случайно или псевдослучайно, когда ГПСЧ должен остановить генерацию псевдослучайных чисел; значение ГПСЧ после того, как он остановился, выбирается в качестве псевдослучайного числа для использования.

73. Эрик Р. Гердс, Роджер Дж. Джонстон и Джеймс Е. Дойл, «Предлагаемый подход к мониторингу демонтажа ядерных боеголовок», *Наука и всеобщая безопасность* 9 (2) (2001): 18-30.

74. Сжатие данных позволяет иметь даже больше памяти.

75. Bruce Schneier, *Crypto-Gram Newsletter,* June 15, 1998, http://www.schneier.com/crypto-gram-9806.html

76. С точки зрения авторов, идентификация третьей стороной данных международных гарантий, в любом случае, несущественна. Третья сторона, сомневающаяся в сведениях гарантий, не будет (или не должна быть) убеждена в правдивости, основанной только на видимой целостности хранимой или переданной информации. Вероятность того. что аппаратура мониторинга была вскрыта или работает неправильно, или же просто была неверно интерпретирована, всегда будет вызывать опасения, которые исключительно трудно смягчить – особенно в умах сторон, психологически не желающих допустить возможности нарушения договора.