M.A. Styugin

The Methods of Information Security Based on Blurring of System

Keywords: information security, protection from research, perfect secrecy, pseudo random number, cipher.

The paper present the model of researching system with own known input, output and set of discrete internal states. These theoretical objects like an absolutely protected from research system and an absolutely indiscernible data transfer channel are defined. Generalization of the principle of Shannon Secrecy are made. The method of system blurring is defined. Theoretically cryptographically strong of absolutely indiscernible data transfer channel is proved and its practical unbreakable against unreliable pseudo random number generator is shown. This paper present system with blurring of channel named Pseudo IDTC and shown asymptotic complexity of break this system compare with AES and GOST.

М.А. Стюгин

МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ, ОСНОВАННЫЕ НА РАЗМЫВАНИИ СИСТЕМ[1]

Введение

Ключевым аспектом данной работы является формализация задачи защиты от исследования системы, представленной в виде совокупности входов, выходов и дискретного внутреннего состояния. Такая формализация позволяет нам ввести идеальные объекты, такие как абсолютно защищенная от исследования система, а также обосновать применимость некоторых технических решений в области информационной безопасности.

Ранее уже была проделана работа по формализации условий, в которых может находиться исследователь относительно объекта исследования [1]. Допустим, исследуемый объект имеет функцию *f(par)*, связывающую его входы и выходы. Под значение *par* мы подразумеваем множество всех возможных аргументов функции, т.е. $f(p_1, \ldots, p_n)$. Входы объекта могут также быть неизвестными параметрами, то есть текущее множество входов объекта выбирается из некоего множества $<par> \in \{<par>\}$. Здесь и далее под фигурными скобками будем понимать все возможные значения данной переменной. В данном случае это множество всех возможных сочетаний *<par>*, которые исследователь может рассматривать как входы объекта. Множество всех параметров на входе может делиться как на множество наблюдаемых для исследователя параметров $\{par\}^{\mathbf{v}}$, так и на множество ненаблюдаемых $\{par\}^{\mathbf{nv}}$:

$$\{par\} = \{par\}^{\mathbf{v}} \cup \{par\}^{\mathbf{vn}}, \{par\}^{\mathbf{v}} \cap \{par\}^{\mathbf{vn}} = \varnothing.$$

Аналогично и множество возможных выходов объекта:

$$\{f\} = \{f\}^{\mathbf{v}} \cup \{f\}^{\mathbf{vn}}, \{f\}^{\mathbf{v}} \cap \{f\}^{\mathbf{vn}} = \varnothing.$$

В результате множество условий, в которых может находиться исследователь по отношению к объекту, можно выразить кортежем из трех значений:

$$\{<par>' = <par>, par \in \{par\}^{\mathbf{v}}, \mathrm{f} \in \{f\}^{\mathbf{v}}\}.$$

[1] Работа поддержана грантом Президента Российской Федерации МК-5025.2016.9.

Здесь <par>′ – набор переменных, которое исследователь выбрал как «вход» объекта; *par* – текущее состояние входа объекта; *f* – текущее состояние выхода объекта. Данная запись немного отличается от приведенной в [1]. В таком виде она более удобна для цепочки дальнейших выводов. В приведенном кортеже есть три условия: исследователь может неправильно выбрать множество параметров, которое является входом объекта <par>′. Истинное множество параметров может не входить во множество его параметрической видимости ($\{par\}^{v}$), а выход объекта может не входить во множество его функциональной видимости ($\{f\}^{v}$). Самый «плохое» состояние исследуемой системы для исследователя может быть выражено кортежем

$$\{<par>' \neq <par>, par \notin \{par\}^{v}, f \notin \{f\}^{v}\}.$$

В таком случае исследователь даже не может поставить задачу исследования, что было показано в [1].

Для дальнейшей формализации добавим в кортеж дополнительные параметры:

$$\{f'(par') = f(par), f(par) \in \{f(par)\}^{v}, <par>' = <par>, par \in \{par\}^{v}, <f>' = <f>, f \in \{f\}^{v}\}.$$

Равенство $f'(par') = f(par)$ свидетельствует о том, что исследователь правильно определил функцию $f'(par')$ исследуемого объекта $f(par)$. Аналогично равенство <*f*>′ = <*f*> показывает, что исследователь правильно выбрал выход <*f*>′ объекта. Условие $f(par) \in \{f(par)\}^{v}$ свидетельствует о том, что функция объекта является наблюдаемой.

Заметим также, что, с практической точки зрения, нам, как правило, нет смысла в разделении факта видимости тех или иных переменных и знания их значений. Если переменная наблюдаема, значение ее известно. В результате оставим в кортеже только принадлежность к видимым элементам:

$$\{par \in \{par\}^{v}, f(par) \in \{f(par)\}^{v}, f \in \{f\}^{v}\}.$$

Для более удобной записи будем обозначать как 1 выполнение условие (принадлежит) и как 0 – невыполнение (не принадлежит). В результате идеальная для исследователя система записывается кортежем (1, 1, 1), что характеризует систему с выполненными условиями $\{par \in \{par\}^{v}, f(par) \in \{f(par)\}^{v}, f \in \{f\}^{v}\}$. Для начала определим значение выражения «решить задачу исследования». Решить задачу исследования – это значит привести систему к состоянию (1, 1, 1). Следовательно, защищенная от исследования система не может быть приведена исследователем в состояние (1, 1, 1). Заметим также, что в реальной системе видимость двух любых параметров всегда означает видимость третьего. Например, мы знаем переменную на входе и знаем функцию системы, таким образом, мы можем вычислить значение на выходе.

В результате, защищенная от исследования система может существовать только в состояниях с одной видимой переменной. Такие состояния имею следующую структуру:

$$(0, 0, 1) - \{par \notin \{par\}^{v}, f(par) \notin \{f(par)\}^{v}, f \in \{f\}^{v}\};$$
$$(0, 1, 0) - \{par \notin \{par\}^{v}, f(par) \in \{f(par)\}^{v}, f \notin \{f\}^{v}\};$$
$$(1, 0, 0) - \{par \in \{par\}^{v}, f(par) \notin \{f(par)\}^{v}, f \notin \{f\}^{v}\}.$$

Определение 1. *Защищенная от исследования система* – это такая система, в которой знание (видимость) одной из характеристик (значение входа, значение выхода,

функция зависимости выхода от входа) не позволяют вычислить две другие характеристики системы.

Абсолютно защищенная от исследования система

По каждому из условий мы можем выразить абсолютно защищенной от исследования системы по аналогии с существующим в шифровании принципом ShannonSecrecy [2].

Определение 2.1. *Абсолютно защищенная от исследования система в состоянии (0, 0, 1)* это система, для которой соблюдаются следующие условия

$$\begin{aligned}&\Pr\left(f=\{f\}\right)=\Pr\left(par=\{par\}\middle|f=\{f\}\right);\\&\Pr\left(f=\{f\}\right)=\Pr\left(par=\{par\}=\{f(par)\}\middle|f=\{f\}\right).\end{aligned}\tag{1}$$

Определение 2.2. *Абсолютно защищенная от исследования система в состоянии (0, 1, 0)* это система, для которой соблюдаются условия

$$\begin{aligned}&\Pr\left(f(par)=\{f(par)\}\right)=\Pr\left(par=\{par\}\middle|f(par)=\{f(par)\}\right);\\&\Pr\left(f(par)=\{f(par)\}\right)=\Pr\left(f=\{f\}\middle|f(par)=\{f(par)\}\right).\end{aligned}\tag{2}$$

Определение 2.3. *Абсолютно защищенная от исследования система в состоянии (1, 0, 0)* это система, для которой соблюдаются условия

$$\begin{aligned}&\Pr\left(par=\{par\}\right)=\Pr\left(f=\{f\}\middle|par=\{par\}\right);\\&\Pr\left(par=\{par\}\right)=\Pr\left(f(par)=\{f(par)\}\middle|par=\{par\}\right).\end{aligned}\tag{3}$$

Далее определим при каких условиях система может находиться в состояниях (1), (2) и (3). Обозначим как $S(1) = 1$ выполнение условий (1) и как $S(1) = 0$ – невыполнение. Аналогично $S(2)$ и $S(3)$.

Теорема 1.1. $S(1) = 1 \Rightarrow |\{f(par)\}| \geq \min(|\{par\}|, |\{f\}|)$.

То есть система может находиться в состоянии (1) тогда, когда множество различных функций между входом и выходом системы не меньше, чем размер множества значений выхода или входа, в зависимости от того, что меньше.

Теорема 1.2. $S(2) = 1 \Rightarrow |\{par\}| \geq |\{f\}|$.

Данная теорема говорит о том, что система может находиться в состоянии (2) тогда, когда размер множества возможных вариантов входа системы больше либо равен размеру множества значений выхода системы.

Теорема 1.3. $S(3) = 1 \Rightarrow |\{f(par)\}| \geq |\{f\}|$

Данная теорема говорит о том, что система в состоянии (3) должна иметь размер множества возможных выходов не менее, чем размер множества возможных функций системы.

Все приведенные условия являются необходимыми, но не достаточными для построения абсолютно защищенной от исследования системы.

В соответствии с принятой в литературе терминологией будем обозначать как «uniform» абсолютно случайную функцию или последовательность бит, в которой каж-

дый последующий бит может быть предсказан не более чем с вероятностью ½ вне зависимости от асимптотической сложности вычислений.

Далее, определим понятие uniform-функции:

$$f(par) \quad \text{un iform} \quad \forall par: \Pr(f) = 1/|\{f\}|.$$

То есть функция является uniform, если выбрана таким образом, что для исследователя любое значение функции на выходе равновероятно. То есть сама функция детерминированная, а не вероятностная. Но при этом она выбрана случайным способом, не дающим какой-либо информации исследователю.

Теорема 2.1. $S(1) = 1 \quad |\{f(par)\}| \geq \min(|\{par\}|, |\{f\}|)$ и $f(par)$ un iform

Для доказательства этой теоремы рассмотрим невыполнение первой части условия:

$$|\{f(par)\}| < \min(|\{par\}|, |\{f\}|).$$

При этом для конкретно взятого значения f, которое нам известно по условию $S(1)$, мы не можем для каждого параметра *par*найти такую функцию, что $f(par) = f$, поскольку $|\{f(par)\}| < |\{par\}|$. Таким образом, мы можем сократить множество$\{par\}$ убрав оттуда неиспользуемые значения. В результате, условие $\Pr(f = \{f\}) = \Pr(par = \{par\} \mid f = \{f\})$ не выполняется. Второе условие теоремы также свидетельствует о том, что если функция используемая в системе не является uniform, то это приводит к нарушению обеих условий: $\Pr(f = \{f\}) = \Pr(par = \{par\} \mid f = \{f\})$ и $\Pr(f = \{f\}) = \Pr(f(par) = \{f(par)\} \mid f = \{f\})$.

Теорема 2.2. $S(2) = 1 \quad |\{par\}| \geq |\{f\}|$ и par un iform

Аналогично доказательству теоремы 2.1.

Теорема 2.3. $S(3) = 1 \quad |\{f(par)\}| \geq |\{f\}|$ и $f(par)$ un iform

Аналогично доказательству теоремы 2.1.

Построение абсолютно защищенной от исследования системы

Рассмотрим множество функций системы $\{f(par)\}$. В реальных системах, как правило, представляется невозможным сделать абсолютно функционально неизвестную техническую систему, поскольку мы ограничены протоколами, стандартами, технологиями и пр. В результате мы можем выделить некий класс систем, характеризующийся функцией $f_k(par)$, где область определения k характеризует варианты представления данной функции. Если k – битовая переменная и $\|k\|$ – ее побитовая длинная, то количество функций $f_k(par)$ определяется как $2^{\|k\|}$.

В соответствии с определенной выше абсолютно защищенной от исследования системой попробуем рассмотреть конкретный пример ее реализации. Классической защищенной от исследования системой является устройство шифрования/дешифрования информации. В первом случае мы имеем доступ к выходу устройства, по которому мы не должны определить его функцию и значение на входе, то есть это устройство (0, 0, 1). Второе устройство имеет известное значение на входе, по которому нельзя получить какую-либо информацию о значении на выходе или функции устройства, то есть это устройство класса (1, 0, 0). Первое устройство определено функцией $f_1(par_1)$, второе – функцией $f_2(par_2)$, как показано на рис. 1. Так же, как было определено выше, $|\{f_1(par_1)\}| = \|k_1\|$ и $|\{f_2(par_2)\}| = \|k_2\|$.

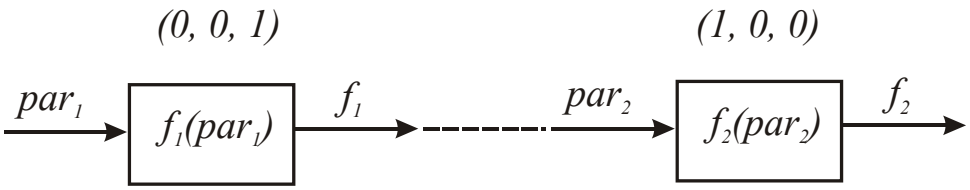

Рис. 1. Устройство шифрования/дешифрования информации

Задача шифрования и дешифрования информации накладывает дополнительные условия на систему:

1) поскольку на входе и выходе у нас одинаковые сообщения, то $\|par_1\| = \|f_2\|$;

2) поскольку информация, в общем случае, не может быть восстановлена из меньшего количества информации, то $\|par_1\| \leq \|f_1\|$.

В соответствии с теоремой 2.1: $|\{f_1(par_1)\}| \geq \min(|\{par_1\}|, |\{f_1\}|)$ и $f_1(par_1)$ un iformОтсюда следует, что

$\|k_1\| \geq \min(\|par_1\|, \|f_1\|)$ $\|k_1\| \geq \|par_1\|$,
k_1 – uniform.

В соответствии с теоремой 2.2: $|\{f_2(par_2)\}| \geq |\{f_2\}|$и $f_2(par_2)$ un iformОтсюда следует, что

$\|k_2\| \geq \|f_2\|$ $\|k_2\| \geq \|par_1\|$,
k_2 – uniform.

Таким образом, мы получаем требования, определенные для шифра Вернама (one-timepad). Это доказывает, что шифр Вернама является не только реализацией шифрования из абсолютно защищенных от исследования модулях в состоянии (0, 0, 1) и (1, 0, 0), но также и то, что данная реализация – единственно возможны.

Релевантной такой постановке задачи является также проблема генерации случайных чисел. На обычном компьютере мы не можем получить достаточную длину последовательности, близкой к случайной. В результате у нас есть некое ограниченное количество бит *k*, являющееся uniform и закладывающееся как стартовое состояние алгоритма (seed) в функцию генерации псевдослучайной последовательности *f* большей длины, чем *k*. Поскольку *f* больше чем *k*, то условие теоремы 2.1 не соблюдается. Однако мы можем взять функцию$f_k(par)$ для которой $\|k\| \geq \|f\|$ и сделать ее мультипликацию. В результате получим следующую функцию:

$$f(par) = f(f_k(par)^1, f_k(par)^2, \dots, f_k(par)^m) = f_k(par)^m,$$

для которой

$$|\{f(par)\}| = m\|k\|.$$

Техническая реализация 1 (Multi Pseudo Random Number Generator – M-PRNG). Рассмотрим следующую техническую систему. Есть серверный модуль *S*, задачей которого является генерация случайных чисел криптографического качества. Данный модуль использует для генерации случайных чисел псевдослучайный алгоритм

$f_k(par)^S$ с начальным состоянием алгоритма (seed) длины $\|k\|$. Также он задействует инфраструктуру сторонних хостов в количестве h, каждый из которых также использует seed длины $\|k\|$ и псевдослучайный алгоритм $f_k(par)$. В начале сервер псевдослучайно выбирает любой хост в системе и отправляет ему запрос. Выбранный хост генерирует псевдослучайное число *f* и псевдослучайное число следующего хоста в цепочке. Следующий хост также генерирует псевдослучайное число и объединяет его операцией XOR с предыдущим $hash(f)$. И так далее, пока цепочка не превысит опционально заданную длину. Полученное в цепочке число *f* возвращается на сервер.

Определение 3. Последовательность псевдослучайных чисел, полученная на выходе генератора *G*, абсолютно непредсказуема на длине *n* бит, если путем исчерпывающего перебора всех возможных входных параметров алгоритма мы не можем детерминировать дальнейшую последовательность алгоритма, начиная с $(n+1)$-го бита.

Теорема M-PRNG. Генератор псевдослучайных чисел, описанный в Технической реализации 1, абсолютно непредсказуем на длине последовательности $\|f\| \leq \|k\| + 1$.

Данную теорему представляется затруднительным доказать теоретическим путем, поскольку она использует понятие «псевдослучайная функция». Однако система M-PRNG была реализована и протестирована на реальных данных. Практическим путем удалось доказать истинность теоремы M-PRNG на рекомендованных псевдослучайных алгоритмах NIST и длинах *k* меньше 10 бит, где условия проверялись путем исчерпывающего перебора всех возможных значений *seeds* и полученного значения функции. Отсюда представляется возможным расширить выводы на большую длину *seed*.

Абсолютная непредсказуемость практическим путем доказывается следующим образом. Если мы на выходе генератора получаем 2^m различных неповторяющихся последовательностей бит, то на любой длине менее *m* бит данный генератор абсолютно непредсказуем.

Описанная задача очень важна с практической точки зрения, так как генерация случайных чисел – ключевая проблема криптографии. Отдельно взятый компьютер не способен сгенерировать длинное и непредсказуемое случайное число. Однако инфраструктура из нескольких компьютеров может справиться с этой задачей даже если, как в примере M-PRNG, компьютеры не являются доверенными. Отдельно взятое случайное число на каждом отдельном хосте в цепочке не позволяет как-либо спрогнозировать полученное псевдослучайное число *f*.

Другой вывод, который можно сделать из описанной технической системы: *функциональное пространство* $|\{f(par)\}|$ *технической системы из нескольких компьютеров ограниченно суммой длины случайных чисел, которые можно получить на каждом хосте*. Это условие – важное ограничение построения абсолютно защищенных от исследования технических систем.

Размывание характеристик системы

Ранее было определено, что абсолютно защищенная от исследования система может существовать только с одним видимым параметром. В практических задачах такой невидимости добиться не всегда возможно. Однако это не означает, что в этих условиях создать абсолютно защищенную от исследования систему невозможно. Добиться этого возможно путем «размывания» видимого параметра. Рассмотрим простой пример. В предложенной на рис. 1 схеме шифратора/дешифратора потенциальный злоумышленник узнал внутренний ключ. Таким образом, система (0, 0, 1) → (1, 0, 0) перешла в состояние (0, 1, 1) → (1, 1, 0). Система стала небезопасной, потому что значение ее входа и выхода теперь может быть легко вычислено.

Систему (0, 1, 1) → (1, 1, 0) мы можем перевести в состояние (0, 1, 0) → (0, 1, 0) путем «размывания» значений входа и выхода. Тем самым мы вновь делаем систему абсолютно защищенной от исследования. Чтобы понять смысл слова «размывание», представим, что мы имеем систему с неизвестными входами. То есть можем наблюдать некую среду, в которой находится система, но не можем определить, какие из параметров этой среды система использует как «вход».

Множество возможных состояний входа, по которому можно передать входные параметры *par*, обозначим как <*par*>.

Обозначим далее $\cup par$ совокупное состояние всех возможных входов объекта $< par >$:

$$< par >_1, \dots, < par >_n \in \{< par >\};$$
$$\bigcup_{i=1\dots n} < par >_i = \{< par >\};$$
$$\cup par = (par_1, \dots, par_n);$$
$$\cup \{par\} = \{(par_1, \dots, par_n)\};$$

$\cup par$ – это и есть *среда*, используемая объектом в качестве входа, а $\cup \{par\}$ – множество состояний, в которых эта среда может находиться.

Если исследователь наблюдает среду параметров входа, а не сам вход, то будем обозначать это следующим образом:

$$(0_1, \dots, \dots).$$

Продолжая эту аналогию, мы можем наблюдать не среду параметров входа, а среду, в которой находится среда параметров входа

$$(0_{0_1}, \dots, \dots)$$

с соответствующими характеристиками $\{<< par >>\}$, $\cup\cup par$ и $\cup\cup \{par\}$. И т. д.

В некоторых случаях состояние $(0_{\dots_1}, \dots, \dots)$ будет эквивалентно состоянию (0, …, …).

Определение 4.1. $\left(0_{\underbrace{\dots_1}_{n}}, \dots, \dots\right) = (0, \dots, \dots)$, если $\Pr\left(\underbrace{\cup \dots \cup}_{n} par = \underbrace{\cup \dots \cup}_{n} \{par\}\right) =$

$$= \Pr\left(par = \{par\} \mid \underbrace{\cup \dots \cup}_{n} par = \underbrace{\cup \dots \cup}_{n} \{par\}\right).$$

Теорема 3.1. $\left(0_{\underbrace{\dots_1}_{n}}, \dots, \dots\right) = (0, \dots, \dots)$ $\quad |\cup \{par\}| \quad \dots \quad \left|\underbrace{\cup \dots \cup}_{n} \{par\}\right| \geq |\{par\}|.$

Данная теорема говорит о том, что система является абсолютно защищенной, если количество вариаций среды, к которой имеет доступ исследователь, больше или равно количеству вариаций самих параметров.

Аналогично можно определить $\cup f$ – совокупное состояние всех возможных выходов объекта $< f >$:

$$< f >_1, \dots, < f >_n \in \{< f >\};$$
$$\bigcup_{i=1\dots n} < f >_i = \{< f >\};$$
$$\cup f = (f_1, \dots, f_n);$$
$$\cup \{f\} = \{(f_1, \dots, f_n)\}.$$

Определение 4.2. $\left(\dots,\dots,0_{\underbrace{\dots 1}_{n}}\right)=(\dots,\dots,0)$, если $\Pr\left(\underbrace{\cup\dots\cup}_{n}f=\underbrace{\cup\dots\cup}_{n}\{f\}\right)=$

$=\Pr\left(f=\{f\}\,|\,\underbrace{\cup\dots\cup}_{n}f=\underbrace{\cup\dots\cup}_{n}\{f\}\right)$.

Теорема 3.2. $\left(\dots,\dots,0_{\underbrace{\dots 1}_{n}}\right)=(\dots,\dots,0)$ $\quad|\cup\{f\}|\ \dots\ \left|\underbrace{\cup\dots\cup}_{n}\{f\}\right|\geq|\{f\}|$.

Аналогично можно определить $\cup f(par)$ – совокупное состояние всех возможных функций объекта $<f(par)>$:

$$<f(par)>_1,\dots,<f(par)>_n\in\{<f(par)>\};$$

$$\bigcup_{i=1\dots n}<f(par)>_i=\{<f(par)>\};$$

$$\cup f(par)=(f(par)_1,\dots,f(par)_n);$$

$$\cup\{f\}=\{(f(par)_1,\dots,f(par)_n)\}.$$

Определение 4.3. $\left(\dots,0_{\underbrace{\dots 1}_{n}},\dots\right)=(\dots,0,\dots)$, если $\Pr\left(\underbrace{\cup\dots\cup}_{n}f(par)=\right.$

$\left.=\underbrace{\cup\dots\cup}_{n}\{f(par)\}\right)=\Pr\left(f(par)=\{f(par)\}\,|\,\underbrace{\cup\dots\cup}_{n}f(par)=\underbrace{\cup\dots\cup}_{n}\{f(par)\}\right)$.

Теорема 3.3. $\left(\dots,0_{\underbrace{\dots 1}_{n}},\dots\right)=(\dots,0,\dots)$ $\quad|\cup\{f(par)\}|\ \dots\ \left|\underbrace{\cup\dots\cup}_{n}\{f(par)\}\right|\geq$

$\geq|\{f(par)\}|$.

Пример построения системы с размытыми характеристиками

Рассмотрим техническую реализацию описанного в предыдущем параграфе примера. По аналогии с абсолютным шифром (one-timepad) такую систему можно ассоциировать с *абсолютно неразличимым каналом передачи информации*. Раннее в работе [3] уже была представлена техническая реализация абсолютно неразличимого канала использующего параметры среды профиля пользователей в социальных сетях. Здесь мы более формально обозначим условия существования таких каналов и их отличие от задач стеганографии.

Техническая реализация 2 (Indiscernible Data Transfer Channel – IDTC). Рассмотрим некую переменную $c\in C$, представленную в виде последовательности 0 и 1. Данная переменная будет характеризовать состояние среды объекта. В состоянии среды объекта мы можем закодировать некую информацию. Есть некий ключ *k*, побитовая длина которого равна побитовой длине состояния среды *c*: $\|k\|=\|c\|$. Ключ k – uniform. Есть некое исходное сообщение *m*, $\|m\|=\|k\|/2$. Исходное сообщение вложено в *c* следующим образом: если в *k* текущий бит равен 0, то текущий бит *c* остается без изменений; если в *k* текущий бит равен 1, то текущий бит в *c* заменяется битом из *m*. В результате мы получаем новое состояние среды $c'\in C$, из которого можно извлечь сообщение *m*, зная ключ *k*.

Легко показать, что описанная техническая реализация удовлетворяет условиям теорем 3.1 и 3.2:

$(0,1,0_1)=(0,1,0\,)$, поскольку $|\cup\{f_1\}|\geq|\{f_1\}|$;

$(0_1,1,0\,)=(0,1,0\,)$, поскольку $|\cup\{par_2\}|\geq|\{par_2\}|$.

В свою очередь объект (0, 1, 0) удовлетворяет условию теоремы 2.2, поскольку $|\{f_1\}|=|\{par_1\}|$ и $|\{f_2\}|=|\{par_2\}|$. На рис. 2 приведена данная реализация.

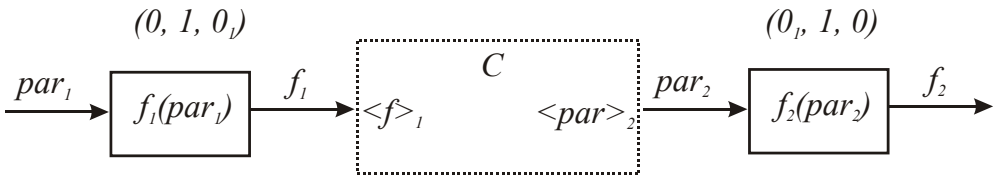

Рис. 2. Абсолютно неразличимый канал передачи информации

Таким образом, получена абсолютно защищенная от исследования система. Принципиальным отличием от постановки задачи стеганографии и мер, используемых для оценки ее эффективности [4], является то, что факт передачи информации был, но никакой анализ данных среды (контейнера *c*) не дает возможности определить формат передачи информации (переменная $< par >$). Однако описанная выше техническая реализация не исключает возможности решения, в том числе, и задачи стеганографии, если мы накладываем дополнительные условия на состояние среды и сообщение *m*.

Приведенная в предыдущем разделе техническая реализация абсолютно защищенной от исследования системы (IDTC) сходна по заявленным характеристикам с системой one-timepad, но при этом имеет ключ в два раза более длинный, чем абсолютный шифр. Может возникнуть вопрос практической целесообразности построения такой технической реализации.

Приведенная система с размытыми параметрами имеет практическую ценность ввиду сложности реализации абсолютных шифров на практике. Здесь можно выделить два преимущества использования такого подхода:

1) абсолютно неразличимый канал передачи информации, как и любые другие размытые параметры, можно использовать в качестве дополнительного рубежа защиты в совокупности с шифрованием либо иным сокрытием информации (структуры, параметров и пр.);

2) применение только лишь одного метода размытия параметров с ненадежным pseudorandom на практике (а не uniform, как в теории), обеспечивает большую криптостойкость шифра.

Рассмотрим первое утверждение. Использование схемы $(0, 1, 0_1)$ $(0_1, 1, 0)$ не исключает использование схемы $(0, 0, 0_1)$ $(0_1, 0, 0)$. В последнем случае мы имеем два рубежа защиты. То есть информация сначала шифруется на неизвестном ключе (преобразуется неизвестной функцией, если дословно читать схему), а затем вкладывается в среду на неизвестном ключе (аналогично – неизвестной функцией). К среде потенциальный исследователь имеет доступ, непосредственно к каналу передачи информации – нет. Отсюда проведена аналогия с абсолютно неразличимым каналом передачи информации. Аналогично мы можем и далее размывать среду, создавая всё новые и новые рубежи защиты:

$$(0, 0, 0_1) \quad (0_1, 0, 0) \quad (0, 0, 0_{0_1}) \quad (0_{0_1}, 0, 0) \quad \left(0, 0, 0_{0_{0_1}}\right) \quad \left(0_{0_{0_1}}, 0, 0\right)$$

Второе утверждение подразумевает сложность криптоанализа системы в случае, если мы используем ненадежный uniform. Такую систему можно представить как гибрид двух приведенных выше технических реализаций M-PRNG и IDTC. Упрощенно будем называть ее PseudoIDTC. Схема PseudoIDTC изображена на рис. 3.

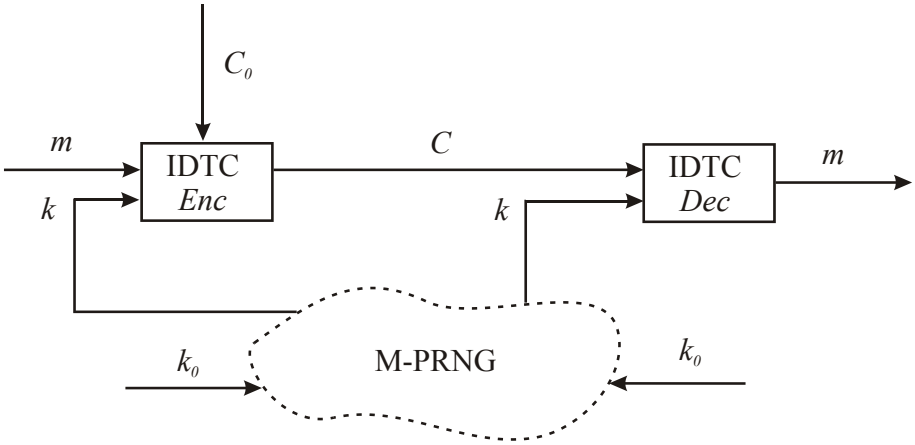

Рис. 3. Схема работы PseudoIDTC

В схеме PseudoIDTC мы принудительно задаем *seed* для генератора псевдослучайных чисел – это ключ k_0. Длина k_0 определяется длиной абсолютно непредсказуемой последовательности бит M-PRNG в соответствии с определением 3. На выходе получаем ключ k, длина которого в битах равна длине среды C_0. Далее, в соответствии с ранее описанной схемой IDTC, сообщение m закладывается в среду C_0 с использованием ключа k. Аналогично происходит дешифрование.

Криптоанализ системы PseudoIDTC

В случае с pseudoone-timepad (использующего pseudo-random вместо uniform [5]), мы можем на основе знания алгоритма генерации pseudo-random расшифровать текст. Такие методы, известные достаточно давно, описаны, например, в [6]. Однако и на современных алгоритмах генерации pseudo-random можно реализовать аналогичные атаки (например, с использованием backdoor, как в случае с Dual_EC_DRBG [7]). Подобные атаки имеют следующий общий алгоритм:

1. Формирование гипотезы относительно какой-либо семантической составляющей текста (например, наличие определенного слова в тексте или предложения)
2. Формирование гипотезы относительно нахождения семантики в тексте.
3. Вычисление последовательности псевдослучайных бит, равной по длине семантическому предложению.
4. Вычисление внутренних параметров pseudo random number generator (PRNG).
5. Нахождение всей последовательности pseudo random и вычисление всего текста.
6. Проверка на результат. Результат удовлетворительный? Если нет – переходим на шаг 2. Если вся длина текста исчерпана – переходим на шаг 1.

Если не принимать во внимание сложность шага 1, который основывается на внемодельных соображениях, весь алгоритм имеет линейную сложность от длины исходного сообщения (шаг 2). Если мы не сталкиваемся с экспоненциальной сложностью в решении задачи на шаге 4 (как в приведенных примерах из [6] и [7]), то текст будет раскрыт.

Рассмотрим тот же самый алгоритм с учетом использование схемы PseudoIDTC:

1. Формирование гипотезы относительно какой-либо семантической составляющей текста (например, наличие определенного слова в тексте или предложения).
2. Формирование гипотезы относительно нахождения семантики в среде.
3. Формирование гипотезы относительно распределения семантики на данной позиции в среде.

4. Вычисление последовательности псевдослучайных бит, равной по длине семантическому предложению.
5. Вычисление внутренних параметров pseudo random number generator (PRNG).
6. Нахождение всей последовательности pseudo random и вычисление всего текста.
7. Проверка на результат. Результат удовлетворительный? Если нет – переходим на шаг 3. Если все варианты распределение перебраны – переходим на шаг 2. Если вся длина текста исчерпана – переходим на шаг 1.

Здесь длина перебираемой последовательности равна не длине исходного текста, как в предыдущем случае, а длине среды. Перебор осуществляется не по байтам, а по битам. То есть сложность шага 2 в этом алгоритме в 16 раз больше, чем в предыдущем, хотя и остается линейной.

Принципиальноу отличие – появление в этой схеме дополнительного шага 3. Допустим, у нас есть некая битовая последовательность (среда) *C*, и мы предполагаем, что в этой последовательности зашифрован некий текст, элементом которого является битовая последовательность *m* (семантическая составляющая). Далее, предполагаем, что данная битовая последовательность зашифрована начиная с первого бита. Сколько различных вариантов ключа *k* мы получим по этой гипотезе? Анализ можно проводить путем построения дерева вариантов ключа (рис. 4).

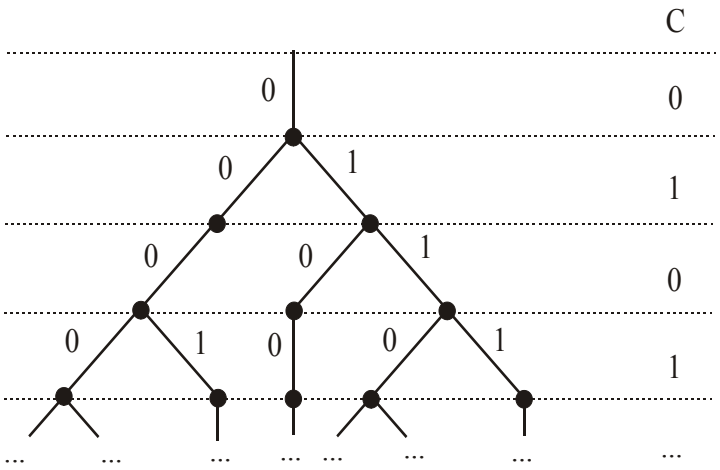

Рис. 4. Построение дерева вариантов ключа

На рис. 4 показано построение дерева возможных вариантов ключа начиная с первого бита контейнера, *C* = 0101100… и сообщения *m* = 1010… На каждой подветке дерево может иметь либо 0 либо 0 или 1. Единица невозможна в тех случаях, когда в контейнере *C* текущий бит не совпадает с текущем битом *m*. В результате на каждом уровне каждая ветка дерева раздваивается, если $m(cur) = C(cur)$, и не раздваивается в случае $m(cur) \neq C(cur)$. Получается, что с вероятностью 0,5 количество веток удваивается и с вероятностью 0,5 остается прежним. В результате математическое ожидание искомого числа веток на каждом уровне – это геометрическая прогрессия с коэффициентом 3/2. Если *n* – текущая глубина дерева, то количество веток на последнем уровне равно $(3/2)^n$. Необходимо также заметить, что минимально необходимая глубина дере-

ва – это гарантированно непредсказуемая длина генератора M-PRNG. Если она равна 128 бит, то количество вариантов ключей, которое мы получим на этом шаге, – $(3/2)^{128}$.

Таким образом, мы получили неполиномиальную сложность вычисления ключа, зависящего от параметра *n*. Можно ли увеличить параметр *n*? Минимально необходимая длина при анализе семантики в битах должна быть не меньше, чем гарантированно непредсказуемая длина генератора PRNG (определение 3). Следовательно, при помощи M-PRNG можно увеличить гарантированно непредсказуемую длину генератора, тем самым увеличив параметр *n*.

В результате даже с использованием ненадежного алгоритма генерации uniform в качестве элементов M-PRNG задача компрометации системы (в отличие от pseudoonetimepad) является вычислительно сложной.

Поскольку техническая реализация PseudoIDTC сходна со схемой шифрования на симметричном ключе, то целесообразно ее сравнение с существующими методами симметричного шифрования. Самым популярным и надежным алгоритмом на сегодняшний день считается AES. В сравнении с AES, схема PseudoIDTC имеет следующие преимущества:

1. Схема PseudoIDTC является дополнительным методом защиты на уровне канала передачи информации и не исключает использование любых методов шифрования на уровне самого сообщения, в том числе и AES.

2. В случае использования качественного uniformPseudoIDTC превращается в PerfectCipher в то время как AES всегда имеет асимптотическую сложность вскрытия от длинны ключа.

3. Длина ключа в AES жестко задана и не регулируется, в то время как в PseudoIDTC гарантированно непредсказуемая длина генератора M-PRNG может быть любой.

4. Не доказана устойчивость шифра AES к алгебраическим атакам типа XSL [8], хотя мы предполагали их применимость к алгоритмам PRNG при анализе криптостойкостиPseudoIDTC.

Вычислительная нагрузка на систему алгоритма AES в 5–8 раз выше, чем при шифровании той же последовательности при помощи PseudoIDTC, и может быть разнесена при помощи M-PRNG на сторонние недоверенные хосты.

Аналогичные преимущества можно обнаружить и в сравнении с таким алгоритмом симметричного шифрования, как российский стандарт симметричного шифрования ГОСТ 28147-89. У схемы Pseudo IDTC существует и недостаток, выраженный в том, что шифрованный текст (минимально необходимый размер среды) в два раза больше исходного.

Выводы

В данной работе удалось сформулировать основные положения теории абсолютно защищенных от исследования систем. Интересными практическими выводами из данной теории являются такие методы, применимые в области информационной безопасности, как увеличение длины гарантированно невскрываемых операций в кооперации ресурсов, а также метод размывания характеристик технической системы. Дано точное определение ранее уже использованного термина абсолютно неразличимого канала передачи информации [3]. Доказана неполиномиальная сложность компрометации системы с размытыми параметрами, даже если компрометация алгоритмов генерации псевдослучайных чисел имеет линейную сложность.

Также была продемонстрирована техническая реализация неразличимого канала передачи информации PseudoIDTC. Обоснована возможность его применения в совокупности с классическими методами шифрования, а также показано преимущество обособленного использования в сравнении с симметричными алгоритмами шифрования.

СПИСОК ЛИТЕРАТУРЫ:

1. Mikhail Styugin. Protection against System Research // Cybernetics and Systems: An International Journal. Volume 45, Issue 4, 2014
2. Shannon, C. E. A Mathematical Theory of Communication, Bell System Technical Journal, July 1948, p.623
3. Mikhail Styugin. Absolutely Indiscernible Data Transfer Channel // Proceedings of The 14th European Conference on Cyber Warfare and Security (ECCWS-2015).pp. 286-294
4. Christian Cachin. An information-theoretic model for steganography.Information and Computation.Volume 192, Issue 1, Pages 41–56
5. Jonathan Katz and Yehuda Lindell.Introduction to Modern Cryptography.CRC PRESS. p. 498
6. James Reeds. «Cracking» a Random Number Generator.Cryptologia, 1977, Volume 1(1), pp. 20-26
7. Dual_EC_DRBG.https://en.wikipedia.org/wiki/Dual_EC_DRBG
8. Nicolas T. Courtois, Josef Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Volume 2501 of the series Lecture Notes in Computer Science. 08 November 2002, pp 267-287

REFERENCES:

1. Mikhail Styugin. Protection against System Research // Cybernetics and Systems: An International Journal. Volume 45, Issue 4, 2014
2. Shannon, C. E. A Mathematical Theory of Communication, Bell System Technical Journal, July 1948, p.623
3. Mikhail Styugin. Absolutely Indiscernible Data Transfer Channel // Proceedings of The 14th European Conference on Cyber Warfare and Security (ECCWS-2015).pp. 286-294
4. Christian Cachin. An information-theoretic model for steganography.Information and Computation.Volume 192, Issue 1, Pages 41–56
5. Jonathan Katz and Yehuda Lindell.Introduction to Modern Cryptography.CRC PRESS. p. 498
6. James Reeds. «Cracking» a Random Number Generator.Cryptologia, 1977, Volume 1(1), pp. 20-26
7. Dual_EC_DRBG.https://en.wikipedia.org/wiki/Dual_EC_DRBG
8. Nicolas T. Courtois, Josef Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Volume 2501 of the series Lecture Notes in Computer Science. 08 November 2002, pp. 267-287
Summary