V. A. Kiryukhin, On the security aspects of protocol CRISP, *Mat. Vopr. Kriptogr.*, 2024, Volume 15, Issue 1, 57–81

DOI: 10.4213/mvk462

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ
2024 Т. 15 № 1 С. 57–81

УДК 519.719.2 DOI https://doi.org/10.4213/mvk462

О безопасности протокола CRISP*

В. А. Кирюхин

ООО «СФБ Лаб», АО «ИнфоТеКС», Москва

Получено 01.IX.2023

Аннотация. С использованием доказательного подхода к оценке свойств криптосистем представлено обоснование стойкости российского стандартизированного протокола CRISP, предназначенного для проверки целостности, обеспечения конфиденциальности и защиты от повторного навязывания сообщений. Протокол рассматривался в качестве специфической схемы аутентифицированного шифрования с ассоциированными данными (AEAD), при этом учитывалась возможность использования одного ключа разными отправителями и в разных криптонаборах. Предъявлены требования к используемым криптонаборам, показано, что существующие криптонаборы соответствуют им, а стойкость протокола в этом случае сводится к стойкости шифра «Магма». На основе полученных оценок даны рекомендации по нагрузке на ключ.

Ключевые слова: CRISP, доказуемая стойкость, AEAD, Магма

On the security aspects of protocol CRISP

V. A. Kiryukhin

LLC «SFB Lab», JSC «InfoTeCS», Moscow

Abstract. Using the provable security approach, we analyze CRISP – a standardized Russian cryptographic protocol that aims to ensure confidentiality, integrity of transmitted messages, as well as protection against replay attacks. The main features of the protocol are non-interactivity, multicasting, and dynamic selection of a cipher suite. The protocol is considered as a specific mode of authenticated encryption with associated data (AEAD). We take into account that one key can be used by many protocol's participants and in different cipher suites. We impose requirements for the set of the cipher suites used in the protocol and show that the existing ones meet them. The security of the protocol is reduced to the PRF-security of KDF and to the security of AEAD-algorithms in all cipher suites. For the protocol with existing cipher suites, only the PRP-security of the «Magma» cipher is required. We obtain heuristic estimates for this computational problem using existing attacks on «Magma». Estimates of the maximum

* Статья предоставлена Организационным комитетом симпозиума CTCrypt'2023.

Citation: Matematicheskie Voprosy Kriptografii, 2024, v. 15, № 1, pp. 57–81 (Russian)

allowable amount of data processed using a single key are also given for existing cipher suites.

Keywords: CRISP, provable security, AEAD, Magma

1. Введение

CRISP (CRyptographic Industrial Security Protocol) [4, 5] — неинтерактивный протокол защищенной передачи данных, разработанный для применения в индустриальных системах. Целевые свойства безопасности, которые должны обеспечиваться протоколом – целостность и (опционально) конфиденциальность сообщений, а также защита от навязывания повторных сообщений.

К важным особенностям протокола CRISP относятся следующие.

Неинтерактивность. Получатель и отправитель не устанавливают сессию, используются предварительно распределенные ключи. Каждое сообщение содержит всю необходимую для обработки информацию. Сообщения могут приниматься в произвольном порядке.

Широковещательность. Одно и то же сообщение одного отправителя может быть предназначено многим получателям. Все пользователи информационной системы могут обладать одним и тем же базовым секретным ключом.

Динамический выбор криптонабора. Отправитель для каждого сообщения может выбирать любой криптонабор из доступных (см. п. 4.1), при этом одни криптонаборы предусматривают шифрование и имитозащиту, а другие только имитозащиту.

В настоящей работе предпринята попытка формального обоснования криптографических свойств протокола за счет использования подходов к *доказуемой оценке стойкости* криптосистем [9, 10]. Рассматриваемая формальная модель учитывает декларируемые свойства безопасности и вышеупомянутые особенности протокола.

Отсутствие механизмов аутентифицированной выработки общего ключа (AKE) делает бессмысленным применение формальных моделей типа CK (Canetti – Krawczyk) [13, 14]. Программы для формальной верификации, такие как AVISPA [12], здесь также практически бесполезны по тем же причинам. Подчеркнем, что представленные доказательства (опирающиеся на подход Рогавэя и Беллара) дают не только качественные, но и что более важно, *количественные* характеристики, учитывающие «неидеальность» используемых криптопримитивов, тогда как инструменты формальной верификации предполагают без-

условную «абсолютную» стойкость режимов имитозащиты и шифрования и позволяют получать только качественные оценки стойкости протокола.

Изложение результатов проводится по следующей схеме. В разделе 2 вводятся необходимые обозначения и определения, приводятся краткие сведения о доказательном подходе к обоснованию стойкости. В третьем разделе описан протокол CRISP.

Четвертый раздел посвящен общему анализу стойкости протокола. Неформально описаны возможности и цели противника, оговариваются ограничения математических моделей, приводятся требования к используемым криптонаборам. Демонстрируется, что ввиду относительной простоты и отсутствия интерактивности CRISP может рассматриваться в качестве режима аутентифицированного шифрования с ассоциированными данными (AEAD). В оговоренных предположениях о применяемой совокупности криптонаборов доказывается стойкость протокола в модели угроз, учитывающей практические возможности противника.

Раздел 5 содержит результаты анализа существующих криптонаборов, использующих блочный шифр «Магма» [1] в режиме имитозащиты CMAC [2] и в режиме гаммирования CTR [2]. Представлены известные оценки стойкости указанных режимов, в том числе композиции «шифрование, затем имитозащита» (encrypt-then-mac). На основе известных методов криптоанализа получены эвристические оценки стойкости для шифра «Магма». Определена допустимая *нагрузка на ключ* (объем данных, который может быть обработан при одном ключе) и упомянуты возможные способы ее увеличения.

2. Обозначения и общие сведения

Будем использовать следующие обозначения: n – битовый размер блока шифра; k – битовый размер ключа шифра; $\tau \leqslant n$ – длина имитовставки; $\oplus$ – побитовое сложение по модулю 2; $||$ – конкатенация двоичных строк; V^* – множество двоичных строк конечной длины; V^n – множество всех n-битных строк; $V^{\leqslant L}$ – множество двоичных строк с длиной не более L бит, включая пустую строку; $|X|$ – битовая длина строки X; $Y \xleftarrow{\text{R}} \mathbf{Y}$ – случайный и равновероятный выбор Y из множества $\mathbf{Y}$; $\mathrm{Func}(\mathbf{X}, \mathbf{Y})$ – множество всех функций, отображающих множество $\mathbf{X}$ в множество $\mathbf{Y}$; $\mathrm{Perm}(\mathbf{X})$ – множество всех биективных преобразований (подстановок) множества $\mathbf{X}$.

Под противником будем понимать интерактивный вероятностный алгоритм $\mathcal{A}$, взаимодействующий с другими алгоритмами (оракулами) с помощью запросов. В рамках произвольной модели угроз TM (threat model) для криптографической схемы (протокола) Alg количественную характеристику возможностей противника $\mathcal{A}$ обозначаем $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$. В зависимости от TM под $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$ обычно понимается преобладание в задаче различения («идеальный» или «реальный») или вероятность реализации некоторой угрозы. Ресурсы противника $\mathcal{A}$ характеризуются числом операций (t) и числом запросов (q) – адаптивно выбираемых пар вход/выход. Без потери общности предполагаем, что $\mathcal{A}$ всегда делает ровно q различных запросов. Размер описания противника $\mathcal{A}$ (его исходный код) ограничен некоторым малым значением. Алгоритм действий оракула (или нескольких оракулов) описывается определением соответствующей модели угроз TM. Результатом вычислений $\mathcal{A}$ после взаимодействия с оракулами $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_w$, $w \in \mathbb{N}$, является значение x (обычно битовое), обозначаем это записью $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_w} \Rightarrow x$.

Максимум значений $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$ среди противников, возможности которых ограничены ресурсами (t, q), обозначаем

$$\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) = \max_{\text{по всем } \mathcal{A} \text{ с ресурсами } (t,q)} \mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A}).$$

В некоторых моделях угроз информационные ресурсы включают в себя числа запросов к разным оракулам, длины этих запросов и т. д. Значение Adv для таких моделей определяется аналогично.

Криптоалгоритм Alg неформально называем стойким в модели угроз TM (TM-стойким), если $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) \leqslant \varepsilon$, где ε не превосходит некоторого малого значения, определяемого требованиями к стойкости, а ресурсы t и q сопоставимы с доступными противнику на практике.

Символ «$\lessapprox$» используется с целью демонстрации практической значимости результатов. Под ним подразумевается: «меньше или равно, если соответствующие эвристические предположения истинны». При этом также может иметь место незначительное округление.

Дадим здесь определения часто используемых формальных моделей, а наиболее важные приведем позднее.

Определение. *Преобладанием противника $\mathcal{A}$ в модели PRP (pseudorandom permutation), атакующего шифр $\mathsf{E} : \mathbf{K} \times \mathbf{X} \to \mathbf{X}$, назовем*

$$\mathrm{Adv}^{PRP}_{\mathsf{E}}(\mathcal{A}) = \left|\mathbf{P}\left(K \xleftarrow{\mathrm{R}} \mathbf{K} \colon \mathcal{A}^{\mathsf{E}_K(\cdot)} \Rightarrow 1\right) - \mathbf{P}\left(\Pi \xleftarrow{\mathrm{R}} \mathrm{Perm}(\mathbf{X}) \colon \mathcal{A}^{\Pi(\cdot)} \Rightarrow 1\right)\right|,$$

$\mathbf{K}$ и $\mathbf{X}$ – множества ключей и блоков соответственно.

Определение. *Преобладанием противника* $\mathcal{A}$ *в модели* PRF (*pseudorandom function*), *атакующего криптоалгоритм* $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$, *назовем*

$$\mathrm{Adv}_{\mathsf{F}}^{PRF}(\mathcal{A}) = \left| \mathbf{P}\left(K \xleftarrow{\mathrm{R}} \mathbf{K} \colon \mathcal{A}^{\mathsf{F}_K(\cdot)} \Rightarrow 1\right) - \mathbf{P}\left(\mathsf{R} \xleftarrow{\mathrm{R}} \mathrm{Func}(\mathbf{X}, \mathbf{Y}) \colon \mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1\right) \right|,$$

$\mathbf{K}, \mathbf{X}, \mathbf{Y}$ – *множества ключей, сообщений и выходов соответственно*.

Определение. *Преобладанием противника* $\mathcal{A}$ *в модели* $VO\text{-}PRF$ (*variable output*), *атакующего криптоалгоритм* $\mathsf{F} \colon \mathbf{K} \times V^* \times \mathbb{N} \to V^*$, *назовем*

$$\mathrm{Adv}_{\mathsf{F}}^{VO\text{-}PRF}(\mathcal{A}) = \left| \mathbf{P}\left(K \xleftarrow{\mathrm{R}} \mathbf{K} \colon \mathcal{A}^{\mathsf{F}_K(\cdot,\cdot)} \Rightarrow 1\right) - \mathbf{P}\left(\mathsf{R} \xleftarrow{\mathrm{R}} \mathrm{Func}(V^* \times \mathbb{N}, V^*) \colon \mathcal{A}^{\mathsf{R}(\cdot,\cdot)} \Rightarrow 1\right) \right|.$$

Запрос от $\mathcal{A}$ *к оракулу имеет вид* $(X, L) \in V^* \times \mathbb{N}$, *где* X – *преобразуемые данные*, L – *битовая длина выхода*.

Определение. *Преобладанием противника* $\mathcal{A}$ *в задаче различения двух криптосистем* S *и* $\widetilde{\mathsf{S}}$ (*с одинаковыми интерфейсами*) *назовем*

$$\mathrm{Adv}_{\mathsf{S},\widetilde{\mathsf{S}}}^{IND}(\mathcal{A}) = \left| \mathbf{P}\left(\mathcal{A}^{\mathsf{S}} \Rightarrow 1\right) - \mathbf{P}\left(\mathcal{A}^{\widetilde{\mathsf{S}}} \Rightarrow 1\right) \right|.$$

Условия, в которых действует противник $\mathcal{A}$ согласно вышеприведенным определениям, соответствуют атакам с адаптивно выбираемыми открытыми текстами (сообщениями).

3. Описание протокола

3.1. Поля сообщения

Сообщение (пакет) протокола CRISP состоит из заголовка, фактического содержимого (`PayloadData`) и имитовставки (`ICV`). Заголовок, в свою очередь, содержит пять полей: `ExternalKeyIdFlag`, `Version`, `CS`, `KeyId`, `SeqNum`. Размерности полей приведены в таблице ниже, суммарная длина всех семи полей не превышает 2048 байтов.

Перечень полей CRISP-сообщения

Номер	Наименование	Обозначение	Длина в битах	
1	`ExternalKeyIdFlag`	–	1	Заголовок H
2	`Version`	–	15	
3	`CS`	CS	8	
4	`KeyId`	–	От 8 до 1024	
5	`SeqNum`	SN	48	
6	`PayloadData`	P и C	Переменная	Содержимое
7	`ICV`	T	Переменная	Имитовставка

Поля `ExternalKeyIdFlag` и `KeyId` позволяют идентифицировать базовый ключ K, используемый для обработки сообщения. Длина поля `KeyId` однозначно определяется первым байтом этого поля.

Нулевой флаг `ExternalKeyIdFlag` $= 0$ означает, что ключ однозначно определяется значением `KeyId`, в противном случае для однозначного определения ключа требуется дополнительная внешняя информация.

Поле `Version` является зафиксированным и зарезервировано для возможных будущих модификаций протокола.

Поле `SeqNum` содержит порядковый номер SN сообщения (счетчик).

Поле `CS` – идентификатор (номер) криптографического набора. В состав криптонабора входят:

– `EncryptionAlg` – алгоритмы Enc и Dec шифрования и расшифрования данных (возможно тривиальные – NULL – шифрование не осуществляется);

– `MACAlg` – алгоритм Mac для выработки τ–битной (`MACLength`) имитовставки T;

– `DeriveIV` – алгоритм DerIv формирования синхропосылки для Enc;

– `DeriveKey` – алгоритм KDF для выработки производных ключей из базового ключа и вспомогательный алгоритм DerIvKDF, определяющий зависимость KDF от битов SN.

Композицию Enc и Mac также будем обозначать AE – аутентифицированное шифрование (с ассоциированными данными).

Поле `PayloadData` содержит исходное сообщение P или шифртекст C, если шифрование предусмотрено используемым криптонабором.

Поле `ICV` содержит значение имитовставки T, длина поля определяется используемым криптонабором. Имитовставка вычисляется для данных, содержащихся в полях 1–6.

3.2. Общие ограничения

Протоколом CRISP предполагается, что отправитель и получатель (получатели) имеют общий предраспределенный базовый ключ K с идентификатором K_{ID}. Каждый отправитель обладает уникальным идентификатором `SourceIdentifier` (обозначаем S_{ID}). Неявно задано инъективное соответствие вида $K_{ID} \to (K, S_{ID})$. Разным K_{ID} может соответствовать один и тот же ключ K, к примеру, один ключ может использоваться разными отправителями. Получатель определяет K_{ID} с помощью полей `ExternalKeyIdFlag` и `KeyId`, а также (опционально) за счет некоторых внешних данных (A_{ext}).

3.3. Инициализация порядкового номера сообщения

Перед началом использования базового ключа K отправитель инициализирует значение счетчика $SN \in [0, 2^{48} - 1]$. При заданных (K, S_{ID}) последовательность значений порядкового номера сообщений SN должна быть строго возрастающей, что подразумевает защиту SN от переполнения. Пусть W является двоичным вектором (исходно нулевым), j-й бит которого устанавливается равным единице, если j-е сообщение (от (K, S_{ID})) было принято. Для каждой пары (K, S_{ID}) получатель инициализирует нижнюю ($\underline{SN}$) и верхнюю ($\overline{SN}$) границы окна принятых сообщений, $\underline{SN} = \overline{SN} = 0$. Получатель непосредственно хранит только часть битов из W – с $\underline{SN}$-го по $\overline{SN}$-й включительно. Размер окна W ограничен предопределенной константой $1 \leqslant Size \leqslant 256$, $(\overline{SN} - \underline{SN}) \leqslant Size$.

3.4. Алгоритм действий отправителя

Отправитель с идентификатором S_{ID} выбирает: базовый ключ K с номером K_{ID}, текст P, криптонабор с номером CS.

1) По K_{ID} определяется порядковый номер SN, его значение увеличивается на 1.

2) Формируются производные ключи: K_{MAC} и (если нужно) K_{ENC},

$$(K_{ENC}, K_{MAC}) = \mathsf{KDF}(K, prms),$$

где конкретный вид $prms$ определяется критонабором и может включать CS, S_{ID}, $\mathsf{DerIvKDF}(SN)$ и другие параметры.

3) Формируется заголовок H (поля 1–5), включающий SN и CS.

4) Если криптонабор предусматривает шифрование, то вычисляется $C = \mathsf{Enc}(K_{ENC}, IV, P)$, $IV = \mathsf{DerIv}(SN)$, в противном случае $C = P$.

5) Вычисляется имитовставка $T = \mathsf{Mac}(K_{MAC}, H||C)$.

6) Отправляется сообщение вида $H||C||T$.

3.5. Алгоритм действий получателя

Для обработки сообщения $H'||C'||T'$ (возможно, искаженного противником) получатель выполняет следующие действия.

1. Если версия протокола или криптонабор, указанные в заголовке H', не поддерживаются, то обработка сообщения прекращается.

2. По полям `KeyId`, `ExternalKeyId` и (возможно) внешним данным A_{ext} определяется K_{ID}, а далее соответствующие ему: ключ K, идентификатор отправителя S_{ID}, границы $(\underline{SN}, \overline{SN})$ окна принятых сообщений W. Если ключ K не определен, то обработка сообщения прекращается.

3. Проверяется значение порядкового номера SN:

– если $SN < \underline{SN}$, то обработка прекращается;

– если SN-й бит W равен единице, то обработка прекращается.

4. Формируются производные ключи: K_{MAC} и (если нужно) K_{ENC}, $(K_{ENC}, K_{MAC}) = \mathsf{KDF}(K, prms)$.

5. Вычисляется имитовставка $T'' = \mathsf{Mac}(K_{MAC}, H'||C')$. Если она не совпадает с принятой ($T'' \neq T'$), то обработка прекращается.

6. Обновляется окно принятых сообщений:

– если $\overline{SN} < SN$, то изменяются $\overline{SN} = SN$ и $\underline{SN} = \min(SN - Size + 1, 0)$;

– SN-й бит в W устанавливается равным единице.

7. Если криптонабор предусматривает шифрование, то вычисляется открытый текст $P' = \mathsf{Dec}(K_{ENC}, IV, C')$, $IV = \mathsf{DerIv}(SN)$, в противном случае $P' = C'$.

4. Общий анализ стойкости

Математически строгое обоснование свойств безопасности какого-либо криптомеханизма возможно исключительно в *формальной модели*, включающей в себя качественные и количественные возможности противника, а также его цели. Несоответствие между теорией (моделью) и практикой служит потенциальным источником угроз. Хорошо известным примером является атака на протокол SSL [16], стойкость которого в недостаточно сильной модели была доказана в [15].

Эти соображения показывают, что следует адекватно учитывать в модели имеющиеся у противника на практике возможности устанавливать противнику наиболее *слабую* цель (цели) из возможных, явно оговаривать ограничения формальной модели.

Начнем с неформального описания возможностей противника. Разумеется, он знает о криптосистеме все, кроме ключей. Может адаптивно выбирать открытый текст P и заголовок H, включая криптонабор CS, идентификатор K_{ID} базового ключа (а значит, и отправителя, определяемого по S_{ID}), порядковый номер SN. Естественное ограничение – пары (S_{ID}, SN) не повторяются, т. е. отправитель не использует один и тот же SN при одном и том же ключе. Атакующий может удалять, изменять порядок и произвольным образом модифицировать любое количество сообщений протокола.

В реальности у противника могут иметься возможности, которые *не* будут учтены в формальной модели: смена версии протокола (предполагается, что существует только одна), компрометация участника протокола (непосредственный доступ к базовому ключу), сведения из побочных каналов (side-channel attacks), сведения в результате сбоев (fault attacks). Стойкость протокола в условиях скомпрометированности производных ключей кратко рассматривается в конце настоящего раздела.

Целями противника являются следующие.

1) Получение какой-либо информации (помимо битовой длины) о защищаемых данных P по шифртекстам C.

2) Навязывание сообщения, ранее не сформированного каким-либо отправителем, любому из получателей.

3) Навязывание сообщения, которое было ранее сформировано каким-либо отправителем, но уже принималось получателем (повторное навязывание).

Неинтерактивность протокола, его особенности и указанные цели противника делают подходящей для анализа CRISP хорошо известную модель угроз NAE (*Nonce-based Authenticated Encryption*) [26], которая схожа с $IND\text{-}CCA3$ [22]. Цель противника в модели NAE – различение пар оракулов («реальных» и «идеальных»). Эта цель слабее любой из перечисленных выше. Противника, умеющего, скажем, эффективно осуществлять навязывание или дешифрование, легко можно преобразовать в алгоритм, решающий задачу различения.

Докажем далее, что даже без хранения «дополнительных данных» протокол CRISP обеспечивает конфиденциальность и целостность (с оговоркой, что отправитель не повторяет SN). Хранение же счетчиков и окна принятых сообщений позволяет обеспечить защиту от повторного навязывания.

4.1. Требования к используемым криптонаборам

Под криптонабором протокола будем понимать четверку алгоритмов

$$\mathsf{CS} = (\mathsf{KDF}, \mathsf{DerIvKDF}, \mathsf{AE}, \mathsf{DerIv}),$$

где AE может быть или композицией алгоритма шифрования Enc и алгоритма имитозащиты Mac, или специальным алгоритмом аутентифицированного шифрования (с ассоциированными данными).

Перечислим достаточные условия для доказательства стойкости.

Пусть базовый ключ K используется в нескольких криптонаборах, тогда все они должны использовать *один и тот же* KDF*. Алгоритм KDF должен быть PRF-стойким (в общем случае, при произвольной длине выхода – VO-PRF). Вход KDF должен включать, по меньшей мере, идентификатор отправителя S_{ID} и номер криптонабора CS. Благодаря этому различные пользователи и разные криптоалгоритмы будут использовать ключи, вычислительно неотличимые от независимых. Некоторые биты порядкового номера сообщения SN (обозначаемые $\mathsf{DerIvKDF}(SN)$) также могут являться частью входа KDF, но для производного ключа не должно быть коллизий синхропосылок: $\mathsf{DerIvKDF}(SN) \neq \mathsf{DerIvKDF}(SN')$ или/и $\mathsf{DerIv}(SN) \neq \mathsf{DerIv}(SN')$ при любой паре $SN \neq SN'$.

Если от криптонабора ожидается обеспечение конфиденциальности и целостности, то AE должен являться стойким детерминированным AEAD-алгоритмом (специальным или за счет композиции), что также формализуется моделью угроз NAE. Это требование предъявляется и в случае, когда защищается только целостность (полагаем при этом длину шифруемых данных равной нулю). Если $\mathsf{AE} = \mathsf{Mac}$, то PRF-стойкости алгоритма Mac достаточно. Алгоритмы имитозащиты, использующие синхропосылки, такие как схема Картера – Вегмана [7] в GCM [11] или UMAC [8], также допустимы при такой формализации.

4.2. Протокол в модели NAE

Рассмотрим протокол CRISP в рамках сценария «много отправителей и один получатель используют один предраспределенный ключ». Считаем для начала, что участники протокола не хранят счетчики и другие вспомогательные данные. Введем необходимые определения.

*Одновременное использование различных KDF (к примеру, CMAC-Магма и HMAC-Стрибог) может не приводить сразу же к эффективной атаке на протокол, но в таком случае сведение в рамках доказательства будет выполняться к малоизученным «экзотическим» базовым задачам, требующим обширного конструктивного криптоанализа.

Определение. *Детерминированным алгоритмом аутентифицированного шифрования с ассоциированными данными* (AEAD) назовем пару алгоритмов

$$\begin{aligned}\mathsf{AE}\colon&\mathbf{K}\times\mathbf{N}\times\mathbf{A}\times\mathbf{P}\to\mathbf{C}\times\mathbf{T},\\ \mathsf{AE}^{-1}\colon&\mathbf{K}\times\mathbf{N}\times\mathbf{A}\times\mathbf{C}\times\mathbf{T}\to\mathbf{P}\cup\{\perp\},\end{aligned}$$

где $\mathbf{K}$, $\mathbf{N}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{C}$, $\mathbf{T}$ – множества ключей, синхропосылок (нонсов – «nonce»), ассоциированных данных, открытых текстов, шифртекстов и имитовставок соответственно. Для любых $(C,T)=\mathsf{AE}(K,N,A,P)$ алгоритм AE^{-1} возвращает $P=\mathsf{AE}^{-1}(K,N,A,C,T)$.

Определение ([26])**.** В модели NAE *преобладанием противника* $\mathcal{A}$, *атакующего* AE, назовем

$$\mathrm{Adv}^{NAE}_{\mathsf{AE}}(\mathcal{A})=\mathbf{P}\left(K\xleftarrow{\mathrm{R}}\mathbf{K}\colon\mathcal{A}^{\mathsf{AE}_K(\cdot,\cdot,\cdot),\mathsf{AE}_K^{-1}(\cdot,\cdot,\cdot,\cdot)}\Rightarrow 1\right)-\mathbf{P}\left(\mathcal{A}^{\$(\cdot,\cdot,\cdot),\perp(\cdot,\cdot,\cdot,\cdot)}\Rightarrow 1\right).$$

Оракул \$ на запрос (N,A,P) возвращает последовательность длины $|P|+ext(P)$, состоящую из реализаций двоичных независимых равновероятных случайных величин («идеальную гамму»). Функция $ext(P)$ вычисляет суммарную длину дополнения и имитовставки. Оракул $\perp$ на любой запрос возвращает символ ошибки «$\perp$». Запросы $\mathcal{A}$ к левому оракулу (AE или \$) не содержат повторяющихся синхропосылок N. Противник $\mathcal{A}$ не посылает правому оракулу (AE^{-1} или $\perp$) запросы (N,A,C,T), где (C,T) были получены от левого оракула в ответ на некоторый запрос (N,A,P) (не осуществляет тривиальной перепосылки). $\mathcal{A}$ делает q запросов к левому оракулу и ν запросов к правому, длина запросов не превосходит l блоков по n бит. Вероятностное пространство модели определяется равновероятным выбором заполнений случайных лент у алгоритма $\mathcal{A}$, у оракулов \$, $\perp$ и ключа K.

Везде далее уникальное значение $N\in\mathbf{N}$ однозначно определяется по ассоциированным данным $A\in\mathbf{A}$: считаем, что $\mathbf{N}$ присутствует в определении неявно. Также полагаем, что алгоритм AE (аналогично AE^{-1}) может быть определен только на некотором *подмножестве* в $\mathbf{A}\times\mathbf{P}$, а не на всем множестве.

Поясним, что оракул \$ соответствует по сути абсолютно стойкому шифру, а оракул $\perp$ – идеальному алгоритму имитозащиты. Попытка навязывания против последнего никогда не завершается успехом (всегда возвращается символ ошибки), тогда как для реального алгоритма (AE_K^{-1}) попытка навязывания иногда может завершиться успехом (наблюдаемым противником), что дает простой критерий различения.

Для протокола CRISP выполняются следующие условия:
– $\mathbf{K} = V^k$ (множество базовых ключей);
– $\mathbf{T} = V^{\leqslant \tau_{\max}}$ (все значения поля `ICV`);
– $\mathbf{P} = \mathbf{C} = V^{\leqslant L_P}$ (`PayloadData`);
– $\mathbf{A} \subseteq \mathbf{A}_{\text{ext}} \times \mathbf{H} \times \mathbf{P}$ (внешние данные, заголовок, поле `PayloadData`).

Внешние данные $A_{\text{ext}} \in \mathbf{A}_{\text{ext}}$ рассматриваются в качестве «виртуального» поля в сообщении. Множество $\mathbf{H} \subset V^{\leqslant L_H}$ содержит все допустимые значения заголовка. Величины L_H и L_P не превосходят длины пакета (без имитовставки), $\tau_{\max}$ – максимальная длина имитовставки среди всех используемых криптонаборов.

Ассоциированные данные A содержат заголовок H, и следовательно, порядковый номер SN и номер криптонабора CS. Совокупности из полей `KeyId`, `ExternalKeyIdFlag` в H и (возможно, пустых) данных $A_{\text{ext}} \in \mathbf{A}_{\text{ext}}$ соответствует пара (K, S_{ID}). Предполагается, что это соответствие *инъективно*†, а значит, изменение внешних данных ведет к изменению ключа и/или S_{ID}.

Длина поля `KeyId` может быть разной, но однозначно определяется первым байтом в самом `KeyId`. Следовательно, разным парам (H, P) соответствуют разные пакеты протокола.

Пара $(S_{ID}, SN) \in \mathbf{N}$ рассматривается в качестве нонса.

Если выбранный криптонабор обеспечивает только целостность, то входом для CRISP является $((A_{\text{ext}}, H, P), \varnothing)$, ассоциированные данные A состоят из внешних данных A_{ext}, заголовка H и содержимого P. Если защищаются конфиденциальность и целостность, то вход – это $((A_{\text{ext}}, H, \varnothing), P)$. Длина входа не превосходит $l \leqslant 2^8$ блоков по 64 бита. Указанные ограничения задают подмножество $\mathbf{A} \times \mathbf{P}$, на котором определен CRISP.

Теорема 1. *В модели NAE для преобладания любого противника, атакующего* CRISP *с криптонаборами из* $\mathbf{CS} = \{\mathsf{CS}_1, \ldots, \mathsf{CS}_c\}$,

$$\mathsf{CS}_i = (\mathsf{KDF}, \mathsf{AE}_i, \mathsf{DerIvKDF}, \mathsf{DerIv}_i),\ i = 1, \ldots, c,$$

справедлива оценка

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t, q, \nu) \leqslant \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t', \kappa) + \sum_{j=1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t', q^{(j)}, \nu^{(j)}),$$

†Предположение соответствует практике. Отсутствие инъективности потребует введения более громоздких определений при анализе, но не приведет к какой-либо фактической уязвимости. Кроме того, внешние данные присутствуют в заголовке лишь «виртуально», что затрудняет потенциальные манипуляции противника.

где $\kappa \leqslant q+\nu$, $\sum_{j=1}^{\kappa} q^{(j)} = q$, $\sum_{j=1}^{\kappa} \nu^{(j)} = \nu$, $\mathsf{AE}^{(j)} \in \{\mathsf{AE}_1, \ldots, \mathsf{AE}_c\}$, $t' = O(t + (q+\nu)l)$, *причем*

1) *вход* KDF *содержит* (*однозначным образом кодирует*) S_{ID}, CS, $\mathsf{DerIvKDF}(SN)$;

2) *для любых* $SN \neq SN'$: $\mathsf{DerIvKDF}(SN) \neq \mathsf{DerIvKDF}(SN')$ *или/и* $\mathsf{DerIv}_i(SN) \neq \mathsf{DerIv}_i(SN')$, $i = 1, \ldots, c$.

Доказательство. Идея доказательства проста. Ограничения на использование KDF (стойкость в модели $VO\text{-}PRF$ и оба условия теоремы) позволяют легко заменить его идеальным примитивом (первое слагаемое в оценке). Благодаря такой замене получим несколько криптосистем, обладающих *независимыми* ключами. Число таких систем обозначаем κ ($1 \leqslant \kappa \leqslant q+\nu$). С помощью техники «гибридного аргумента» заменяем каждую из них на идеальную (что соответствует сумме $\sum_{j=1}^{\kappa}(\ldots)$).

Перейдем к изложению схемы доказательства.

Согласно модели NAE противник имеет доступ к паре оракулов. Оракул шифрования (левый) эмулирует работу всех отправителей, оракул проверки целостности (правый) соответствует единственному получателю. Противник выбирает конкретного отправителя за счет манипулирования ассоциированными данными A в своем запросе к оракулу. Согласно сделанным ранее предположениям произвольное изменение внешних данных A_{ext} и/или полей `KeyId`, `ExternalKeyIdFlag` ведет к изменению (K, S_{ID}). В силу единственности K (в рассматриваемом сценарии) такое изменение равнозначно смене отправителя (S_{ID}).

По условию теоремы все криптонаборы должны использовать один и тот же KDF. Рассмотрим протокол CRISP-I, в котором вместо KDF применяется случайная функция $\mathsf{R} \in \mathrm{Func}(V^* \times \mathbb{N}, V^*)$. Построим алгоритм $\mathcal{B}$, для которого верно неравенство

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP},\mathsf{CRISP\text{-}I}}(\mathcal{A}) \leqslant \mathrm{Adv}^{VO\text{-}PRF}_{\mathsf{KDF}}(\mathcal{B}). \tag{1}$$

Алгоритм $\mathcal{B}$ имитирует для $\mathcal{A}$ один из двух протоколов (CRISP или CRISP-I). Каждый запрос от $\mathcal{A}$ к любому из двух оракулов может потребовать вычисления производных ключей, а значит, $\mathcal{B}$ сделает к своему оракулу (KDF или R соответственно) вплоть до $\kappa \leqslant (q+\nu)$ запросов. Один ответ оракула – производный ключ $K^{(j)}$ для алгоритма $\mathsf{AE}^{(j)}$, $j = 1, \ldots, \kappa$. Таким образом, $\mathcal{B}$ обладает всеми производными ключами и может идеально симулировать для $\mathcal{A}$ работу соответствующего протокола. Результат работы $\mathcal{B}$ равен результату работы $\mathcal{A}$.

CRISP-I содержит не более κ *независимых* друг от друга подсистем, каждая характеризуется своим алгоритмом $\mathsf{AE}^{(j)}$ и ключом $K^{(j)}$. Вместо исходной пары оракулов (соответствующих криптосистеме с зависимыми ключами) фактически используются κ пар независимых оракулов. Согласно условию 1 следующие подсистемы независимы:

– с различными криптонаборами (CS – часть входа KDF);

– с одинаковыми криптонаборами, но с разными отправителями (S_{ID} – также часть входа KDF);

– с одинаковыми криптонаборами и S_{ID}, но с разными $\mathsf{DerIvKDF}(SN)$.

Напомним, что пара (S_{ID}, SN) рассматривается в качестве нонса (неповторяющегося значения) для CRISP и CRISP-I. Различные SN могут соответствовать одному $IV = \mathsf{DerIv}_i(SN)$ при $i = 1, \ldots, c$, но в силу условия 2 внутри любой из κ подсистем значения IV не повторяются. Иными словами, при одном и том же ключе $K^{(j)}$ $(1 \leqslant j \leqslant \kappa)$ для формирования пакетов будут использоваться заведомо разные синхропосылки IV, их число равно $q^{(j)}$.

Противник $\mathcal{A}$ фактически выбирает одну из подсистем для взаимодействия за счет выбора ассоциированных данных A в своем запросе (а точнее, тройки значений – S_{ID}, CS, SN).

В j-й подсистеме $\mathcal{A}$ может сделать $q^{(j)}$ (соответственно $\nu^{(j)}$) запросов к левому (соответственно правому) оракулу. Максимум значения $q^{(j)}$ зависит от числа бит SN, которые не влияют на $\mathsf{DerIvKDF}(SN)$. Все попытки навязывания противник может выполнить при единственном наборе (S_{ID}, CS, SN), следовательно, $\nu^{(j)} \leqslant \nu$. Ограничения на общее число запросов имеют вид $\sum_{j=1}^{\kappa} q^{(j)} = q$ и $\sum_{j=1}^{\kappa} \nu^{(j)} = \nu$.

Благодаря независимости ключей в CRISP-I мы можем использовать «гибридный аргумент». Рассмотрим последовательность протоколов[‡] $\mathsf{CRISP\text{-}I}^{(0)}, \ldots, \mathsf{CRISP\text{-}I}^{(\kappa)}$, где $\mathsf{CRISP\text{-}I}^{(0)} = \mathsf{CRISP\text{-}I}$, а $\mathsf{CRISP\text{-}I}^{(\kappa)}$ является «идеальным» – все κ пар оракулов имеют вид $(\$, \perp)$. В $\mathsf{CRISP\text{-}I}^{(j)}$, $0 < j < \kappa$, все пары оракулов с индексами $1, \ldots, j$ заменены на «идеальные» $(\$, \perp)$, а другие пары $(j+1, \ldots, \kappa)$ – «реальные» алгоритмы.

Если $\mathcal{A}$ может эффективно различать $\mathsf{CRISP\text{-}I}^{(j-1)}$ и $\mathsf{CRISP\text{-}I}^{(j)}$, то существует противник $\mathcal{B}^{(j)}$, который может атаковать $\mathsf{AE}^{(j)}$ в модели NAE с такой же эффективностью. Перед началом взаимодействий $\mathcal{B}^{(j)}$ генерирует ключи $K^{(j')}$ для подсистем с индексами $j' > j$. После этого

[‡]Говоря более формально, можно пронумеровать все возможные тройки $(S_{ID}, CS, \mathsf{DerIvKDF}(SN))$ (пусть их будет $\kappa_{\max}$) и рассмотреть все соответствующие подсистемы. В общем случае, $\kappa_{\max} \geqslant \kappa$, но противник не будет делать запросов к $(\kappa_{\max} - \kappa)$ подсистемам, а следовательно, использование $\kappa_{\max}$ вместо κ не повлияет на результат.

$\mathcal{B}^{(j)}$ обрабатывает запросы от $\mathcal{A}$:

– определяет номер пары оракулов (j') по ассоциированным данным;

– если $j' < j$, то симулирует «идеальные» оракулы (\$ или $\perp$);

– если $j' = j$, то перенаправляет запрос собственному оракулу;

– если $j' > j$, то симулирует «реальные» алгоритмы за счет самостоятельно сформированных ключей $K^{(j')}$.

Если сам $\mathcal{B}^{(j)}$ взаимодействовал с парой «реальных» (соответственно «идеальных») оракулов, то для $\mathcal{A}$ идеально симулировался $\mathsf{CRISP\text{-}I}^{(j-1)}$ (соответственно $\mathsf{CRISP\text{-}I}^{(j)}$). Результат работы $\mathcal{B}^{(j)}$ равен результату работы $\mathcal{A}$.

Преобладание $\mathcal{A}$, таким образом, ограничено сверху:

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP\text{-}I}^{(j-1)},\mathsf{CRISP\text{-}I}^{(j)}}(\mathcal{A}) \leqslant \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(\mathcal{B}^{(j)}),$$

$\mathcal{B}^{(j)}$ делает к левому и правому оракулам $q^{(j)}$ и $\nu^{(j)}$ запросов соответственно. Пользуясь неравенством треугольника, получаем

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP\text{-}I}^{(0)},\mathsf{CRISP\text{-}I}^{(\kappa)}}(\mathcal{A}) \leqslant \sum_{j=1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(\mathcal{B}^{(j)}). \tag{2}$$

В силу произвольности алгоритма $\mathcal{A}$ суммирование оценок (1), (2) дает доказываемое неравенство. □

Следствие. *Пусть криптонаборы при запросах к левому* (*соответственно*, *правому*) *оракулу принадлежат множеству* $\mathbf{CS}_{\mathcal{S}}$ (*соответственно*, $\mathbf{CS}_{\mathcal{R}}$), *тогда*

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t,q,\nu) \leqslant \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t',\kappa) + \sum_{j_{\mathcal{S}}=1}^{\kappa'} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j_{\mathcal{S}})}}(t', q^{(j_{\mathcal{S}})}, 0)$$

$$+ \sum_{j_{\mathcal{SR}}=\kappa'+1}^{\kappa''} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j_{\mathcal{SR}})}}(t', q^{(j_{\mathcal{SR}})}, \nu^{(j_{\mathcal{SR}})}) + \sum_{j_{\mathcal{R}}=\kappa''+1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j_{\mathcal{R}})}}(t', 0, \nu^{(j_{\mathcal{R}})}),$$

где $0 \leqslant \kappa' \leqslant \kappa'' \leqslant \kappa \leqslant q + \nu$, *а* $\mathsf{CS}^{(j_{\mathcal{S}})}$, $\mathsf{CS}^{(j_{\mathcal{SR}})}$, $\mathsf{CS}^{(j_{\mathcal{R}})}$ *принадлежат соответственно* $\mathbf{CS}_{\mathcal{S}} \backslash \mathbf{CS}_{\mathcal{R}}$, $\mathbf{CS}_{\mathcal{S}} \cap \mathbf{CS}_{\mathcal{R}}$, $\mathbf{CS}_{\mathcal{R}} \backslash \mathbf{CS}_{\mathcal{S}}$.

Стойкость протокола CRISP к атакам на конфиденциальность определяется самым «слабым» криптонабором у отправителя ($\mathbf{CS}_{\mathcal{S}}$). Формирование подделок может быть наиболее эффективным как при «общем» криптонаборе из $\mathbf{CS}_{\mathcal{S}} \cap \mathbf{CS}_{\mathcal{R}}$ (за счет использования $q^{(j_{\mathcal{SR}})}$ пакетов, защищаемых на одном ключе), так и при «уязвимом» криптонаборе, который поддерживается только получателем ($\mathbf{CS}_{\mathcal{R}} \backslash \mathbf{CS}_{\mathcal{S}}$).

В последнем случае попытки навязывания будут выполняться по сути «вслепую», что соответствует нулю в $\mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j_{\mathcal{R}})}}(t', 0, \nu^{(j_{\mathcal{R}})})$. Хорошо известный пример такой «уязвимости» – короткая имитовставка, а соответствующая атака – простое угадывание.

Напомним, что приведенные выше результаты описывают случай, когда все участники обладают единственным предраспределенным ключом. Если ключей много, то анализ легко сводится с помощью «гибридного аргумента» к нескольким независимым криптосистемам. Получение же нетривиальных оценок в подобных условиях является предметом дальнейших исследований. Представляется, что это возможно, когда протокол используется только для защиты целостности, а S_{ID} в явном виде включается в состав пакета.

Также отметим, что если получателей больше одного, то это не ведет к качественному изменению результатов анализа. Каждый получатель обрабатывает сообщения независимо от других. Пакет не содержит полей с информацией о получателе, предполагается, что им может быть любой, кто знает базовый ключ. На практике число попыток навязывания ν обычно растет линейно с увеличением числа пользователей системы, поэтому сценарий «много получателей» по сути приводит не к иной модели угроз, а к количественному увеличению ресурсов противника.

4.3. Защита от повторного навязывания

Стойкость протокола CRISP к повторному навязыванию сообщений почти очевидна. В самом деле, каждый получатель хранит окно принятых сообщений W. Если пакет с некоторым порядковым номером SN был принят, то этот факт хранится в W. Сообщения с номерами меньше нижней границы ($SN < \underline{SN}$) отклоняются без рассмотрения. Таким образом, два раза принять пакет с одним и тем же номером невозможно, это не зависит ни от содержимого пакетов, ни от используемых криптоалгоритмов. Формальное доказательство этого факта, тем не менее, довольно объемно, требует введения громоздких определений [23–26], и в силу этого не приводится.

4.4. Стойкость при компрометации ключей

Благодаря PRF-стойкости KDF протокол CRISP продолжает обеспечивать некоторые свойства безопасности в условиях, когда отдельные (производные) ключи становятся известны противнику.

Очевидно, утечка базового ключа K ведет к потере всех свойств безопасности. Учитывая, что базовый ключ всегда является частью состояния, хранимого любым участником, можно говорить об отсутствии стойкости к атакам «чтения назад» (свойства «forward secrecy»).

При раскрытии ключа шифрования K_{ENC} атакующий может дешифровать содержимое не более q' пакетов. Максимальное значение q' зависит от $\mathsf{DerIvKDF}$, т. е. от частоты смены производных ключей.

Компрометация ключа имитозащиты K_{MAC} позволит противнику навязать каждому получателю вплоть до q' пакетов. Ограничение порождено сменой производных ключей, которая обязательно произойдет, так как прием последовательности подделок приведет к увеличению счетчика SN.

При утечке любого количества производных ключей противник не может определить значение еще одного ключа (в том числе базового). Противное означало бы, что KDF не является PRF-стойким.

5. Анализ существующих криптонаборов

Существующая версия протокола CRISP [4,5] содержит четыре «парных» криптонабора (см. таблицу ниже).

CS	Название	Имитозащита	Шифр.	Имитовставка (τ бит)
1	MAGMA-CTR-CMAC	+	+	32
2	MAGMA-NULL-CMAC	+	–	32
3	MAGMA-CTR-CMAC8	+	+	64
4	MAGMA-NULL-CMAC8	+	–	64

Все они используют блочный шифр «Магма» [1] $\mathsf{E} : V^k \times V^n \to V^n$, длина ключа $k = 256$ бит, длина блока $n = 64$ бита.

Для шифрования используется режим гаммирования CTR [2], а для защиты целостности – алгоритм CMAC [2]. Синхропосылкой для режима гаммирования служат 32 младших бита (lsb) счетчика пакетов

$$IV = \mathsf{DerIv}(SN) = \mathsf{lsb}_{n/2}(SN),\ SN \in V^{48}.$$

Формирование производных ключей $\mathsf{KDF} : V^k \times V^{\leqslant L} \times \mathbb{N} \to V^{\leqslant d \cdot n}$

также основано на CMAC,

$$\begin{aligned}\Gamma = \mathsf{KDF}(K, X, d) =& \mathsf{CMAC}(K, \mathsf{byte}(1,1)||X||\mathsf{byte}(n\cdot d, 2))||\\ & \mathsf{CMAC}(K, \mathsf{byte}(2,1)||X||\mathsf{byte}(n\cdot d, 2))||\\ & \dots\\ & \mathsf{CMAC}(K, \mathsf{byte}(d,1)||X||\mathsf{byte}(n\cdot d, 2)),\end{aligned}$$

$\mathsf{byte}(x, j)$ – представление целого числа x в виде строки из j байтов. Производные ключи определяются равенствами

$$K_{MAC}||K_{ENC} = \Gamma,\ d = \frac{2\cdot k}{n} = 8 \text{ при } CS \in \{1, 3\},$$
$$K_{MAC} = \Gamma,\ d = \frac{k}{n} = 4 \quad \text{при } CS \in \{2, 4\}.$$

Входные данные X у KDF содержат, среди прочего: номер криптонабора CS, идентификатор отправителя S_{ID}, 35 старших бит (msb) номера $SN \in V^{48}$, $\mathsf{DerIvKDF}(SN) = \mathsf{msb}_{35}(SN)$.

В рамках KDF длина входных данных для CMAC не превосходит 50 байтов (семи n-битных блоков, $l_{KDF} = 7$). Заметим, что из-за зависимости KDF от 35 бит SN на производном ключе (паре ключей) будет защищено не более $2^{48}/2^{35} = 2^{13}$ пакетов.

5.1. Известные оценки стойкости режимов CTR и CMAC

Приведем для релевантных моделей угроз известные оценки стойкости используемых режимов работы шифра. Напомним: q – число защищаемых сообщений (запросов к оракулу), l – максимальная длина одного сообщения в n-битных блоках.

Стойкость CTR с блочным шифром E в модели $IND\text{-}CPNA$ [33, Appendix A] (неотличимость шифртекста от «идеальной гаммы») – прямое следствие PRP-PRF леммы [17], согласно которой [10]

$$\mathrm{Adv}^{IND\text{-}CPNA}_{\mathsf{CTR}}(t, q, l) \leqslant \mathrm{Adv}^{PRP}_{\mathsf{E}}(t', q\cdot l) + \frac{(q\cdot l)^2}{2^{n+1}},\ t' = t + O(ql).$$

PRF-стойкость алгоритма CMAC исследовалась во многих работах [18–21], приводим результат последней из них.

Теорема ([21, Theorem 3.1]). *Преобладание любого противника, атакующего* CMAC *в модели* PRF, *допускает оценку*[§]

$$\mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t, q, l) \leqslant \mathrm{Adv}^{PRP}_{\mathsf{E}}(t', q\cdot l + 1) + \frac{16\cdot q^2 + q\cdot l^2 + 4\cdot q\cdot l}{2^n} + \epsilon(q, l),$$

[§]Значение $\epsilon(q, l)$ пренебрежимо мало и по причине громоздкости здесь не приводится.

где $t' = t + O(q \cdot l)$, $q \cdot (l+1) \leqslant 2^{n-1}$.

PRF-стойкости CMAC достаточно для VO-PRF-стойкости $\mathsf{KDF}(K, X, d)$. Другими словами, KDF неотличима от случайной функции, когда длины входа и выхода могут не являться постоянными. Пусть запросы противника к KDF имеют вид $(X_1, d_1 \cdot n)$,...,$(X_q, d_q \cdot n)$, и $(X_i, d_i) \neq (X_j, d_j)$, $1 \leqslant i < j \leqslant q$. Легко видеть, что в таких условиях все входы у CMAC являются различными, следовательно,

$$\mathrm{Adv}^{VO\text{-}PRF}_{\mathsf{KDF}}(t, q) \leqslant \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', q \cdot d, l_{KDF} = 7).$$

5.2. Стойкость AEAD-режимов

Хорошо известно, что алгоритмы приведенных в таблице (на стр. 73) криптонаборов 1–4 порождают стойкие режимы аутентифицированного шифрования.

Лемма 1. *Преобладание любого противника в модели* NAE, *атакующего криптоалгоритм*

$$\mathsf{CTR\text{-}CMAC} : \mathbf{K} \times \mathbf{A} \times \mathbf{P} \to \mathbf{C} \times \mathbf{T},$$
$$\mathsf{CTR\text{-}CMAC} : (V^k \times V^k) \times V^{\leqslant l \cdot n} \times V^{\leqslant l \cdot n} \to V^{\leqslant l \cdot n} \times V^{\tau},$$

допускает оценку

$$\mathrm{Adv}^{NAE}_{\mathsf{CTR\text{-}CMAC}}(t, q, \nu) \leqslant \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', q + \nu, l) + \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR}}(t', q, l) + \frac{\nu}{2^{\tau}},$$

$t' = t + O((q + \nu) \cdot l)$.

Запросы противника имеют вид (A, P), $A = H$.

Лемма 2. *Преобладание любого противника в модели* NAE, *атакующего криптоалгоритм*

$$\mathsf{NULL\text{-}CMAC} : \mathbf{K} \times \mathbf{A} \times \mathbf{P} \to \mathbf{C} \times \mathbf{T},$$
$$\mathsf{NULL\text{-}CMAC} : V^k \times V^{\leqslant l \cdot n} \times \varnothing \to \varnothing \times V^{\tau},$$

допускает оценку

$$\mathrm{Adv}^{NAE}_{\mathsf{NULL\text{-}CMAC}}(t, q, \nu) \leqslant \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', q + \nu, l) + \frac{\nu}{2^{\tau}},\ t' = t + O((q + \nu) \cdot l).$$

Запросы противника имеют вид $(A, \varnothing)$, $A = H||P$.

Следовательно, оба криптоалгоритма соответствуют требованиям теоремы 1. Доказательство первой леммы представлено в [33, Appendix B]. Вторая лемма – прямое следствие первой при $\mathbf{P} = \mathbf{C} = \varnothing$.

5.3. Эвристические оценки сложности базовой задачи

Стойкость всех используемых в CRISP криптонаборов сводится к единственной базовой задаче — неотличимости шифра «Магма» от случайной подстановки.

На текущем этапе развития теории сложности для конкретного шифра невозможно получить верхние оценки для преобладания противника в модели PRP. Вместе с тем необходимость формирования практических рекомендаций мотивирует оценивать значение $\mathrm{Adv}^{PRP}_{\text{Магма}}(\mathcal{A})$ *эвристически*, за счет сужения множества *всех* алгоритмов $\mathcal{A}$ с ресурсами (t, q) до множества *известных* к настоящему времени методов конструктивного криптоанализа¶.

От шифра ГОСТ 28147-89 шифр «Магма» отличается зафиксированным «хорошим» набором узлов замен, что обеспечивает стойкость к дифференциальному [27] и линейному [28] методам криптоанализа (см. также описание синтеза шифра 2-ГОСТ [32]).

В [31, Corollary 1] предложен простой алгоритм различения, использующий «симметричные неподвижные точки» и основанный на том, что для случайной подстановки Π над V^n вероятность равенства $\Pi(x||x) = x||x$ при произвольном $x \in V^{n/2}$ составляет порядка 2^{-n}, а для шифра «Магма» – вдвое больше $(2 \cdot 2^{-n})$. Следовательно, за счет проверки $q \leqslant 2^{n/2}$ «симметричных точек» преобладание в задаче различения составит $\approx q \cdot (2 \cdot 2^{-n} - 2^{-n}) \approx q \cdot 2^{-n}$.

Различитель можно построить на основе алгоритма определения ключа. Если истинный ключ найден, то ответ различителя – «1» (взаимодействие было с шифром), если нет – «0» (взаимодействие с подстановкой Π). При $q > \frac{k}{n} = 4$ вероятность ложноположительного ответа после взаимодействия с Π близка к нулю. Таким образом, преобладание различителя почти равно вероятности восстановления ключа. Вероятность успеха при тотальном опробовании $t \cdot 2^{-k}$.

Два специальных метода восстановления ключа описаны в [29, 30]. Они устроены схожим образом. Последовательно рассматриваются q пар «открытый текст – шифртекст» (ОТ – ШТ), ОТ выбираются независимо. Для каждой пары с вероятностью 2^{-p} может произойти редкое благоприятное событие. Вероятность реализации хотя бы одного такого события среди q пар ОТ – ШТ не превосходит $q \cdot 2^{-p}$. Для каждой пары в предположении, что событие произошло, выполняется 2^c операций, строится 2^c ключей-кандидатов, каждый из которых проверяется с по-

¶Методы, требующие более 2^k операций на предварительные вычисления (на формирование памяти алгоритма $\mathcal{A}$), при этом исключаются из рассмотрения.

мощью других пар ОТ – ШТ. Если событие действительно произошло, то истинный ключ обязательно будет построен. Общее число опробуемых ключей составит $q \cdot 2^c$.

В рассматриваемой модели противник выполняет t операций (предполагаем, что шифрование блока требует одной операции), а следовательно, при использовании методов [29,30] опробует не все ключи ($q{\cdot}2^c$), а лишь некоторую их долю, не превосходящую $\frac{t}{q\cdot 2^c}$. Вероятность восстановления истинного ключа можно оценить сверху как

$$\frac{q}{2^p} \cdot \frac{t}{q \cdot 2^c} = \frac{t}{2^{p+c}},\ q \leqslant 2^p.$$

Имеет место и другая верхняя оценка – вероятность успеха противника не превосходит вероятности благоприятного события ($q \cdot 2^{-p}$). Таким образом, получаем итоговую верхнюю оценку

$$\min\left(\frac{q}{2^p}, \frac{t}{2^{p+c}}\right).$$

В атаке Исобе [29] использовалось так называемое «свойство отражения» (reflection). Вероятность редкого события при одной паре ОТ – ШТ составляет $2^{-p} = 2^{-\frac{n}{2}} = 2^{-32}$. При каждой паре ОТ – ШТ опробуется $2^c = 2^{192}$ ключей.

В работе [30] Динура, Дункельмана, Шамира применялось свойство «фиксированная точка» (fixed point), $2^{-p} = 2^{-n} = 2^{-64}$, $2^c = 2^{128}$.

Общий вид эвристической оценки таков:

$$\mathrm{Adv}^{PRP}_{\text{Магма}}(t,q) \lessapprox \max_{t_1+t_2+t_3=t}\left(\frac{t_1}{2^{256}}, \min\left(\frac{q}{2^{32}}, \frac{t_2}{2^{224}}\right), \min\left(\frac{q}{2^{64}}, \frac{t_3}{2^{192}}\right)\right) + \min\left(2^{-32}, \frac{q}{2^{64}}\right).$$

Упрощая при $t \ll 2^{192}$ и произвольном $q < 2^{32}$, получаем

$$\mathrm{Adv}^{PRP}_{\text{Магма}}(t,q) \lessapprox \frac{t}{2^{192}} + \frac{q}{2^{64}}.$$

Преувеличивая реальные возможности противника, полагаем, что его вычислительные ресурсы $t \approx 2^{128}$ операций. Оценки стойкости режимов работы шифра (CTR, CMAC) зависят от q квадратично (см. п. 5.1), тогда как оценка PRP-стойкости — лишь линейно. Приведенные соображения позволят далее пренебречь слагаемым $\mathrm{Adv}^{PRP}_{\text{Магма}}$ при численных расчетах.

Схожим образом в эвристической оценке могут быть учтены и другие методы криптоанализа шифра «Магма», которые гипотетически могут появиться в будущем.

5.4. Оценки нагрузки на ключ

Воспользуемся имеющимися оценками преобладания противника, чтобы определить максимально допустимую нагрузку на ключ.

В [3] определены понятия «максимально допустимое значение вероятности однократного навязывания сообщений» (π_{mac}) и «максимально допустимое значение вероятности эффективного применения методов криптографического анализа» (π_{enc}). Модель NAE учитывает атаки как на целостность (попытки навязывания), так и на секретность (к примеру, бесключевое чтение), следовательно, для любого используемого криптоалгоритма Alg должно быть выполнено

$$\mathrm{Adv}^{NAE}_{\mathsf{Alg}}(t, q, \nu) < \pi = \min(\pi_{\text{enc}}, \pi_{\text{mac}}).$$

Для иллюстративных целей выберем $\pi = \min(\pi_{\text{enc}}, \pi_{\text{mac}}) = 2^{-10}$.

Напомним, что κ – число производных ключей, q (соответственно, $q' = 2^{13}$) – число пакетов, защищаемых на одном базовом (соответственно, производном) ключе. Для упрощения расчетов считаем, что благодаря техническим мерам защиты число попыток навязывания ν (соответственно, ν') много меньше числа защищаемых пакетов q (соответственно, q'). Максимальная длина пакета – $l \leqslant \frac{2048 \cdot 8}{n} = 2^8$ блоков. Для выработки одного производного ключа (пары ключей) KDF использует $d \in \{4, 8\}$ обращений к алгоритму CMAC.

Суммируя вышесказанное и упрощая оценки до наиболее значимых слагаемых, получаем (см. п. 5.1):

$$\varepsilon_{KDF} \leqslant \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', \kappa \cdot d = 2^{21} \cdot 8, l_{KDF} = 7) \lessapprox \frac{16 \cdot (\kappa \cdot d)^2}{2^n} = 2^{-12},$$

$$\varepsilon_{CTR} = \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR}}(t', q' + \nu' \approx 2^{13}, l = 2^8) \lessapprox \frac{(q' \cdot l)^2}{2^{n+1}} = 2^{-23},$$

$$\varepsilon_{CMAC} = \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', q' + \nu' \approx 2^{13}, l = 2^8) \lessapprox \frac{16 \cdot (q')^2}{2^n} = 2^{-34}.$$

Для $CS \in \{1, 3\}$ получаем $\varepsilon_{CS} \approx \varepsilon_{CTR} + \varepsilon_{CMAC} \approx \varepsilon_{CTR}$, а для $CS \in \{2, 4\}$ имеет место $\varepsilon_{CS} \approx \varepsilon_{CMAC}$.

Нетрудно видеть, что для каждого криптонабора $\varepsilon_{CS} < \pi$, аналогичное верно для ε_{KDF}, когда число производных ключей $\kappa \leqslant 2^{21}$. Иными словами, если, как обычно, каждый ключ рассматривается *по отдельности*, то при указанных параметрах протокол стойкий.

С другой стороны, если рассмотреть *всю* криптосистему и *все ключи*, то должна быть малой сумма

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t,q,\nu) \leqslant \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t',\kappa)+\kappa\cdot\mathrm{Adv}^{NAE}_{\mathsf{CS}}(t',q',\nu') = \varepsilon_{KDF}+\kappa\cdot\varepsilon_{CS},$$

в которой второе слагаемое имеет наибольшую значимость. Так, для первого криптонабора из требования $(\varepsilon_{KDF}+\kappa\cdot\varepsilon_{CS}) < \pi$ следует существенно более жесткое ограничение $\kappa < 2^{13}$. Кроме того, если заменить производные ключи на истинно случайные, то $(\kappa\cdot\varepsilon_{CS}) < \pi$ и результат *не изменится*. Говоря иначе, KDF и производные ключи не делают CRISP хуже.

Схожий результат может быть получен и конструктивным образом – вероятность того, что атака хотя бы на одну из κ криптосистем окажется успешной, примерно в κ раз больше, чем аналогичная вероятность для *одной* заранее выбранной криптосистемы. Выбор первого (каждый ключ отдельно) или второго (все ключи вместе) подхода должен быть основан на требованиях к конкретной информационной системе.

Отметим, что существует множество способов увеличить допустимую нагрузку на ключ. Наиболее простым и эффективным, пожалуй, является использование шифра с относительно большим размером блока. Для шифра «Кузнечик» [1] длина блока $n = 128$ бит – значение κ увеличивается до недостижимой на практике границы 2^{54}.

Наибольшее влияние на итоговую оценку оказывает значение ε_{CTR}, которое растет квадратично с увеличением числа блоков. Следовательно, увеличить допустимую нагрузку можно за счет: внутренней смены ключей – CTR-ACPKM [6]; усечения выхода шифра до $s < n$ бит (как предусмотрено в [2]); двойного гаммирования с помощью CTR.

6. Заключение

С использованием математического аппарата «доказуемой стойкости» [9, 10] показано, что протокол CRISP [4, 5] обеспечивает свойства целостности и конфиденциальности, а также защиту от повторного навязывания сообщений. Доказательство выполнено за счет сведения к стойкости используемых криптонаборов.

1) криптонаборы, используемые с одним базовым ключом, должны применять один и тот же PRF-стойкий алгоритм выработки производных ключей;

2) применяемые последовательно шифрование и имитозащита должны порождать стойкий детерминированный AEAD-алгоритм.

Существующие криптонаборы [4, 5] удовлетворяют всем требованиям, их стойкость сводится к стойкости шифра «Магма». Полученные результаты позволили дать обоснованные рекомендации по допустимой нагрузке на ключ.

Автор благодарит А. В. Уривского, О. В. Шемякину, А. С. Рыбкина и М. А. Бородина за содержательные дискуссии по теме работы. Огромная благодарность А. А. Щербаченко за ценные замечания и предложения по тексту статьи. Детальные комментарии и список неточностей от рецензентов конференции CTCrypt'2023 помогли значимо улучшить работу – отдельное спасибо!

Список литературы

[1] *ГОСТ Р* 34.12-2015. *Информационная технология. Криптографическая защита информации. Блочные шифры*, М.: Стандартинформ, 2015.

[2] *ГОСТ Р* 34.13-2015. *Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров*, М.: Стандартинформ, 2015.

[3] *Р* 1323565.1.005-2017. *Информационная технология. Криптографическая защита информации. Допустимые объемы материала для обработки на одном ключе при использовании некоторых вариантов режимов работы блочных шифров в соответствии с ГОСТ Р* 34.13-2015, М.: Стандартинформ, 2017.

[4] *Р* 1323565.1.029-2019. *Информационная технология. Криптографическая защита информации. Протокол защищенного обмена для индустриальных систем*, М.: Стандартинформ, 2020.

[5] *ГОСТ Р. Информационная технология. Криптографическая защита информации. Протокол защищенного обмена для индустриальных систем*, М.: Российский институт стандартизации, 2024.

[6] *Изменение №1 ГОСТ* 34.13-2018. *Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров*, М.: ФГБУ «РСТ», 2023.

[7] Wegman M., Carter L., "New hash functions and their use in authentication and set equality", *J. Comput. System Sci.*, **22** (1981), 265–279.

[8] Black J., Halevi S., Krawczyk H., Krovetz T., Rogaway P., "UMAC: fast and secure message authentication", CRYPTO '99, Lect. Notes Comput. Sci., **1666**, 1999, 216–233.

[9] Bellare M., Rogaway P., *Introduction to Modern Cryptography*, Davis: Univ. of California at Davis, 2005.

[10] Rogaway P., *Evaluation of Some Blockcipher Modes of Operation*, CRYPTREC, Unpublished manuscript, 2011.

[11] McGrew D. A., Viega J., "The security and performance of the Galois/Counter Mode (GCM) of operation", INDOCRYPT 2004, Lect. Notes Comput. Sci., **3348**, 2004, 343–355.

[12] Armando A. et al., "The AVISPA tool for the automated validation of internet security protocols and applications", CAV 2005, Lect. Notes Comput. Sci., **3576**, 2005, 281–285.

[13] Canetti R., Krawczyk H., "Analysis of key-exchange protocols and their use for building secure channels", EUROCRYPT 2001, Lect. Notes Comput. Sci., **2045**, 2001, 453–474.

[14] LaMacchia B., Lauter K., Mityagin A., "Stronger security of authenticated key exchange", ProvSec 2007, Lect. Notes Comput. Sci., **4784**, 2007, 1–16.

[15] Krawczyk H., “The order of encryption and authentication for protecting communications (or: how secure is SSL?)”, CRYPTO 2001, Lect. Notes Comput. Sci., **2139**, 2001, 310–331.

[16] Canvel B., Hiltgen A., Vaudenay S., Vuagnoux M., “Password interception in a SSL/TLS channel”, CRYPTO 2003, Lect. Notes Comput. Sci., **2729**, 2003, 583–599.

[17] Chang D., Nandi M., “A short proof of the PRP/PRF Switching Lemma”, *Cryptology ePrint Archive*, *Report* 2008/078, 2008.

[18] Iwata T., Kurosawa K., “OMAC: one-key CBC MAC”, FSE 2003, Lect. Notes Comput. Sci., **2887**, 2003, 129–153.

[19] Iwata T., Kurosawa K., “Stronger security bounds for OMAC, TMAC and XCBC”, INDOCRYPT 2003, Lect. Notes Comput. Sci., **2904**, 2003, 402–415.

[20] Nandi M., “Improved security analysis for OMAC as a pseudorandom function”, *J. Math. Cryptology*, **3**:2 (2009), 133–148.

[21] Chattopadhyay S., Jha A., Nandi M., “Towards tight security bounds for OMAC, XCBC and TMAC”, ASIACRYPT 2022, Lect. Notes Comput. Sci., **13791**, 2022.

[22] Shrimpron T., “A characterization of authenticated-encryption as a form of chosen-ciphertext security”, *Cryptology ePrint Archive*, *Report* 2004/272, 2004.

[23] Kohno T., Palacio A., Black J., “Building secure cryptographic transforms, or how to encrypt and MAC”, *Cryptology ePrint Archive*, *Report* 2003/177, 2003.

[24] Bellare M., Kohno T., Namprempre C., “Breaking and provably repairing the SSH authenticated encryption scheme: a case study of the encode-then-encrypt-and-MAC paradigm”, *ACM Trans. Inf. Syst. Security*, **7**:2 (2004), 206–241.

[25] Boyd C., Hale B., Mjølsnes S. F., Stebila D., “From stateless to stateful: generic authentication and authenticated encryption constructions with application to TLS”, Cryptographers Track at the RSA Conference 2016, Lect. Notes Comput. Sci., **9610**, 2016, 55–71.

[26] Rogaway P., Zhang Y., “Simplifying game-based definitions indistinguishability up to correctness and its application to stateful AE”, CRYPTO 2018, Lect. Notes Comput. Sci., **10992**, 2018, 3–32.

[27] Biham, E., Shamir, A., “Differential cryptanalysis of DES-like cryptosystems”, *J. Cryptology*, 1991, 3–72.

[28] Matsui M., “Linear cryptanalysis method for DES cipher”, EUROCRYPT’93, Lect. Notes Comput. Sci., **765**, 1994, 386–397.

[29] Isobe T., “A single-key attack on the full GOST block cipher”, FSE 2011, Lect. Notes Comput. Sci., **6733**, 2011, 290–305.

[30] Dinur I., Dunkelman O., Shamir A., “Improved attacks on full GOST”, FSE 2012, Lect. Notes Comput. Sci., **7549**, 2012, 9–28.

[31] Kara O., Karakoc F., “Fixed points of special type and cryptanalysis of full GOST”, CANS 2012, Lect. Notes Comput. Sci., **7712**, 2012, 86–97.

[32] Dmukh A. A., Dygin D. M., Marshalko G. B., “A lightweight-friendly modification of GOST block cipher”, *Математические вопросы криптографии*, **5**:2 (2014), 47–55.

[33] Kiryukhin V., “On security aspects of CRISP”, *Cryptology ePrint Archive*, *Report* 2023/1303, 2023.