Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)
КАФЕДРА «Информационная безопасность» (ИУ8)

Курсовая работа
Симметричное шифрование: определение безопасности, конфиденциальности и базовых примитивов

по дисциплине «Криптографические методы защиты информации»

Студент ИУ8-114 (Группа) — Н. В. Железцов (И. О. Фамилия) — _____ (Подпись, дата)

Преподаватель — А. Е. Жуков (И. О. Фамилия) — _____ (Подпись, дата)

Оценка: _____

Москва, 2025 г.

РЕФЕРАТ

В данной курсовой работе исследуются основы симметричных криптосистем с ассоциированными данными и формальные определения безопасности. Работа начинается с расширения функциональности симметричных криптосистем путем введения ассоциированных данных, которые кодируют контекст появления шифротекста. Это позволяет связывать сообщение с его контекстом без включения последнего в само сообщение.

Основное внимание уделяется трем ключевым аспектам безопасности: конфиденциальности, целостности и аутентификации. Подробно анализируются различные подходы к определению конфиденциальности, включая семантическую безопасность, неотличимость (left-or-right безопасность) и безопасность real-or-random. Доказывается эквивалентность этих понятий, что позволяет выбирать наиболее удобное из них для анализа конкретных криптосистем.

Рассматриваются атаки с выбранным открытым текстом (CPA) и выбранным шифротекстом (CCA), а также формализуются соответствующие игры безопасности. Особое внимание уделяется целостности шифротекста и открытого текста, демонстрируется их неравносильность и важность целостности шифротекста для доказательств безопасности.

В работе также представлены базовые криптографические примитивы, такие как потоковые шифры, псевдослучайные функции (PRF) и псевдослучайные перестановки (PRP). Исследуются их свойства и взаимосвязи, включая лемму о переключении между PRP и PRF. Приводятся практические конструкции, такие как режим счётчика (CTR) и режим сцепления блоков (CBC), и анализируется их безопасность [1].

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Эта глава будет посвящена проблеме определения безопасности для симметричной криптографии и соответствующих примитивов, таких как блочные шифры и генераторы потока ключей, а также вопросу, как рассуждать о безопасности различных конструкций, рассмотренных в Главе 1.

Общий подход к анализу криптографических конструкций заключается в том, чтобы построить редукцию, которая использует противника против одной конструкции для выполнения другой задачи. Затем проводится анализ редукции, чтобы определить её вероятность успеха в этой задаче относительно вероятности успеха исходного противника.

Это может выглядеть как замена одной неизвестной величины другой неизвестной величиной. Но это поверхностное впечатление. Основная проблема заключается в том, что, хотя криптографические конструкции могут быть довольно сложными, контекст, в котором требуется их анализировать, будет ещё более сложным. Цель состоит в том, чтобы свести анализ сложных криптографических конструкций в сложных контекстах к анализу более простых криптографических конструкций или даже естественных математических задач в более простых контекстах. В конечном итоге можно использовать методы, набросанные в Главах 1–5, чтобы оценить, насколько трудно атаковать примитивы, и посредством редукций установить оценку сложности атаки криптографической конструкции.

Основная задача этой главы — проанализировать широкий спектр целей безопасности, что является удивительно сложной темой. Затем обсуждаются конструкции и базовые примитивы. Одно важное общее утверждение позволяет рассматривать конфиденциальность и целостность отдельно.

Вторая задача — дать вводное обсуждение каналов. Это в некотором смысле изначальная криптографическая тема: как двум сторонам защитить разговор? К этому вопросу будет возвращение позже.

Хеш-функции были впервые представлены в Разделе 4.2. Строго говоря, хеш-функции не связаны с шифрованием и дешифрованием, но данный концепт обсуждается в этой главе, поскольку проектирование распространённых хеш-функций в некотором смысле связано с проектированием симметричных примитивов. Также рассматриваются идеальные модели для хеш-функций и других примитивов.

О языке. Современный стиль в криптографии состоит в том, чтобы точно определять, что представляет собой противник для схемы и как измерять, насколько хорошо этот противник действует. Таким образом, точно определяется лишь понятие уровня небезопасности, и доказываются соотношения между уровнями небезопасности.

Можно было бы определить уровни безопасности как отрицание уровня небезопасности и формулировать теоремы в терминах уровней безопасности. Однако поскольку фактическое доказательство связывает именно уровни небезопасности, такой подход быстро становится запутанным и неудобным.

Вместо этого принято соглашение, что утверждение о том, что нечто является безопасным, носит неформальный характер, что уместно, поскольку почти все положительные утверждения о уровнях безопасности являются предположениями. Соотношения между уровнями небезопасности затем превращаются в противоположные неформальные утверждения о неформальных заявлениях безопасности. Читателю предлагается сравнить неформальные утверждения в ранних главах с соответствующими теоремами в более поздних главах.

ОСНОВНАЯ ЧАСТЬ

1 Определение безопасности

Прежде чем начинать определение безопасности, сначала будет расширена функциональность симметричных криптосистем. Шифротексты довольно часто не обрабатываются в полной изоляции, но появляются в некотором контексте. Часто неудобно включать этот контекст в сообщение, которое шифруется, но часто было бы удобно как-то вовлекать контекст в шифрование и расшифрование шифротекста, чтобы связать сообщение с его контекстом. На самом деле, отсутствие связи расшифрования шифротекстов с их правильным контекстом является постоянным источником ошибок в проектировании безопасных систем. Контекст кодируется как ассоциированные данные.

D

Определение 1.1

Симметричная криптосистема с ассоциированными данными состоит из множества $\mathcal{K}$ ключей, множества $\mathcal{P}$ открытых текстов, множества $\mathcal{F}$ ассоциированных данных, множества $\mathcal{C}$ шифротекстов и двух алгоритмов:

- алгоритма шифрования E, который по входу ключа, ассоциированных данных и открытого текста выводит шифротекст; и
- алгоритма расшифрования D, который по входу ключа, ассоциированных данных и шифротекста выводит либо открытый текст, либо специальный символ $\perp$ (обозначающий некорректный шифротекст).

Для любого ключа k, ассоциированных данных ad и любого открытого текста m выполняется

$$D(k, ad, E(k, ad, m)) = m.$$

Ассоциированные данные имеют значение только для целостности, а не для конфиденциальности от пассивных наблюдателей. Будет разработан и проанализирован ряд криптосистем, которые не поддерживают ассоциированные данные. Технически множество F можно рассматривать как одноэлементное, но для упрощения изложения ассоциированные данные в этих случаях игнорируются.

1.1 Конфиденциальность

Смысл конфиденциальности должен заключаться в том, что противник не может узнать ничего о содержимом шифротекста. Чтобы это сделать, необходимо точно определить, что означает «узнать что-то». Нужно также как-то определить контекст, в котором это происходит. Далее будут определены несколько вариантов конфиденциальности: один будет определять то, что интуитивно понимается под конфиденциальностью, один будет практичен для работы, и один будет удобен при использовании симметричной криптографии в других конструкциях.

Прежде чем начинать, перечислим два принципа определения безопасности.

- Мы хотим разрабатывать криптосистемы, пригодные для всех. Поэтому мы не знаем точно, какая безопасность понадобится пользователям криптосистемы. Следовательно, требуется разрабатывать настолько широко полезную безопасность, насколько возможно.
- Мы хотим настолько сильную безопасность, насколько возможно при разумных затратах. Безопасность определяется в терминах того, чего противник не должен быть способен сделать, поэтому более сильная безопасность достигается тем, что работа противника упрощается. Нужно быть осторожным, чтобы не сделать работу противника тривиальной.

Довольно длинное обсуждение. Ниже приводится довольно длинное обсуждение, объясняющее одну линию рассуждений, которая в конечном итоге приводит к полезному определению конфиденциальности для симметричных криптосистем. Это включено для того, чтобы дать представление о процессах мышления, лежащих за современными определениями безопасности.

Игра. Один полезный способ определить безопасность — сформулировать её в терминах игры между противником и сущностью, которая в некотором смысле играет роль честных пользователей криптосистемы. Противник отправляет запросы сущности, и сущность отвечает. В некотором смысле проводится эксперимент, в котором противник — испытуемый, поэтому сущность, играющая роль честных пользователей, называется *экспериментом*. Этот эксперимент используется для определения безопасности.

Для заданного определения безопасности противник является переменной величиной, в то время как эксперимент фиксирован. Слово «эксперимент» может включать противника, тогда как слово «игра» всегда относится к разговору между экспериментом и противником.

Нужна игра, в которой эксперимент создаёт шифротекст и отдаёт его противнику, который затем узнаёт что-то о расшифровании шифротекста и сообщает эксперименту, что он узнал.

Цель противника. Что должен узнать противник? Одна возможность — что противник должен узнать полное расшифрование шифротекста. Однако это явно слишком сильно, поскольку существует много примеров из реального мира, где частичное восстановление открытого текста было полезно для противника. Есть даже ситуации, где не было нужно даже частичного восстановления открытого текста, а утекала какая-то другая полезная информация об открытом тексте.

Любая информация, которую хочет узнать противник, может быть закодирована в целое число, поэтому для любого противника существует функция $f : P \to \mathbb{Z}$, определяющая, что противник хочет узнать.

Понять, какое из многих возможных значений является правильным, труднее, чем понять, какое из немногих значений является правильным. Это подсказывает, что противнику было бы выгоднее, если бы функция имела меньший образ, например $\{0, 1\}$.

Нужно, чтобы эксперимент создавал шифротекст, и затем противник должен правильно ответить на один вопрос вида «да/нет» о расшифровании.

Длина сообщения. Теперь следует обсудить одно фундаментальное препятствие: шифротекст не может быть короче сообщения в среднем. Если требуется скрывать длину сообщения, шифротекст должен быть по крайней мере столь же длинным, как самый длинный открытый текст, что может быть крайне дорого. Другими словами, скрывать длину сообщения дорого.

Длину можно частично скрыть, добавляя заполнение случайной длины, или заполняя сообщение так, чтобы длина делилась на фиксированное число, но поскольку это обычно увеличивает длину сообщения, это увеличит стоимость криптосистемы. Поскольку неизвестно, что именно потребуется пользователям

криптосистемы, это подсказывает, что скрывать длину сообщения — не работа разработчика криптосистемы. Пользователь, скорее всего, будет иметь больше информации о компромиссе между стоимостью и пользой и сможет выполнять более разумное заполнение.

Поэтому попыток скрывать длину сообщения предприниматься не будет. Это означает, что будут интересовать только такие функции f, значение которых нельзя вывести из длины сообщения. Это требование трудно формализовать. Однако оно окажется легко разрешимым, поэтому оно пока игнорируется.

Выбор сообщения. Безопасность определяется в терминах игры между экспериментом, который создаёт шифротекст, и противником.

Эксперимент должен зашифровать сообщение, но как он должен его выбирать? Одна возможность — выбирать сообщение из некоторого фиксированного вероятностного распределения. С этим связано несколько проблем.

Прежде всего, криптосистема, вероятно, не будет использоваться таким образом. Сообщения, отправляемые реальными людьми, редко случайны. Хотя любое интересное сообщение будет в некоторой степени непредсказуемым, противник часто будет иметь частичные сведения о сообщении ещё до создания шифротекста.

Также то, как используется криптосистема, может влиять на её безопасность, и исторически это часто было так. Как отмечалось выше, разработчики криптосистемы не могут легко предсказать, как будет использоваться криптосистема. На самом деле, хотелось бы захватывать возможность того, что криптосистема может использоваться таким образом, который помогает противнику.

В идеале хотелось бы, чтобы сообщение выбирал противник, но это очевидно не работает. Вместо этого противнику будет позволено выбирать способ выбора сообщения. Технически противник передаёт эксперименту алгоритм X для выборки из множества открытых текстов.

Алгоритм выборки противника должен быть в некотором смысле корректным. Во-первых, каждый выбранный открытый текст должен иметь одинаковую длину, что аккуратно решает проблему того, что большинство криптосистем раскрывают длину открытого текста. Второе требование связывает функцию f с алгоритмом выборки X.

Цель противника, снова. Хотелось бы, чтобы криптосистема была безопасной независимо от того, какой вопрос пытается ответить противник. Однако очень трудно одновременно квантифицировать по всем возможным вопросам и что-то доказать. Лучший выбор — просто позволить противнику выбрать вопрос.

Иными словами, противник должен решить как выбирается сообщение (предоставляя алгоритм выборки X) и какой вопрос (функцию f) нужно ответить о сообщении. Следует рассматривать алгоритм выборки X как предшествующее знание противника о сообщении, а f — как то, что противник хочет узнать, наблюдая шифротекст.

Напомним, что нужно делать работу противника как можно проще, не делая её тривиальной. Чтобы работа противника не была тривиальной, требуется, чтобы алгоритм выборки X выбирал сообщение m так, что $f(m)$ может быть и 0, и 1. На этом этапе будет введено строгое требование: когда m выбирается алгоритмом X, вероятность того, что $f(m) = 0$, должна быть $1/2$.

Ассоциированные данные. Симметричные криптосистемы были определены так, чтобы включать ассоциированные данные. Идея состояла в том, что ассоциированные данные кодируют контекст, в котором появляется шифротекст. Это будет полезно для проектировщиков систем.

Обычно контекст является открытым, что означает отсутствие необходимости защищать конфиденциальность ассоциированных данных. Однако иногда было бы полезно обеспечивать конфиденциальность ассоциированных данных, поскольку это дало бы разработчикам систем ещё более полезную функциональность. Рассмотрение ассоциированных данных как секретных создаёт ряд технических проблем в изложении и даёт небольшую пользу, поэтому ассоциированные данные считаются открытыми.

Итоги на данный момент. Безопасность определяется как игра между экспериментом и противником. Противник выбирает ассоциированные данные, распределение сообщений и вопрос для ответа, два последних задаются алгоритмом выборки X и функцией $f : P \to \{0,1\}$. Эксперимент выбирает ключ k, выбирает сообщение $m \xleftarrow{r} X$, шифрует сообщение $c \leftarrow E(k, ad, m)$ и отправляет испытательный шифротекст противнику. Противник отвечает предположением $b' \in \{0,1\}$. Говорят, что противник выигрывает игру, если $b' = f(m)$.

Можно заметить, что любой противник может выигрывать с вероятностью $1/2$, просто всегда угадывая 0. Это означает, что интересны только противники, выигрывающие существенно чаще половины случаев. Противники, выигрывающие существенно реже чем половину, тоже интересны, так же как интересен человек, который стабильно предсказывает неправильный результат подбрасывания монеты.

Теперь будет обсуждён альтернативный способ определения этого. Иногда вычисление функции f является сложным — тогда выиграл ли противник? Такие функции можно использовать, при условии что X выбирает одновременно и m, и $f(m)$. Тогда эксперимент выбирал бы $(m, b) \xleftarrow{r} X$, и можно сравнивать b' с b.

В этом определении функция f исчезает. Вместо этого эксперимент по сути выбирает сообщение из двух различных распределений выборки, и задача противника — угадать, из какого распределения было выбрано сообщение. Выбранное распределение задаётся выбранным битом b, который называется *испытательным битом*. В некоторых случаях удобно, чтобы противник предоставлял два различных алгоритма выборки.

Это также позволяет ослабить требования к игре. Ответ на вопрос «да/нет» уже не должен определяться самим сообщением. Очевидно, если одно и то же сообщение может быть выбрано из обоих распределений, работа противника усложняется, поскольку противник уже не может быть прав с вероятностью 1. Но ослабление требований таким образом делает другие задачи проще, что будет видно далее.

Больше испытательных шифротекстов. На практике противник может быть заинтересован не в одном сообщении, а в нескольких сообщениях. В этом случае требуется, чтобы алгоритм выборки сообщений выбирал последовательность сообщений, и сообщения не обязаны быть независимыми. Количество выбранных сообщений и их длина должны быть независимыми от испытательного бита, выводимого в конце.

Однако сообщения могут зависеть не только друг от друга, но и от других факторов, таких как шифротексты. На этом этапе моделирование становится более сложным. Вместо того чтобы приводить один алгоритм выборки нескольких сообщений, противник должен предоставлять последовательность алгоритмов выборки одного сообщения. Каждый алгоритм должен выводить сообщение и некоторое состояние. Эксперимент выполняет алгоритм выборки,

шифрует выбранное сообщение и отдаёт шифротекст противнику. Состояние, выводимое алгоритмом выборки, скрыто от противника. Вместо этого следующий алгоритм выборки получает состояние как вход перед выборкой. Последний алгоритм выборки должен также выводить испытательный бит.

Приведённая идея довольно проста, но гарантировать, что противник не сможет тривиально выиграть, несколько сложнее. Очевидно, противник должен гарантировать, что количество выбранных сообщений и их длина будут независимыми от испытательного бита. Также хотелось бы, чтобы испытательный бит был независим от шифротекстов, созданных экспериментом. Поскольку шифротексты могут быть скоррелированы с сообщениями, а сообщения — со шифротекстами, это слишком сильное требование. Вместо этого нужно гарантировать, что испытательный бит выбирается таким образом, что он не зависит от наблюдаемых шифротекстов. Единственный разумный вариант — выбирать испытательный бит до создания каких-либо шифротекстов.

Замечание. Это действительно исключает некоторые стратегии противника (например, когда испытательный бит является чётностью всех выбранных сообщений), но, похоже, трудно отличить такие стратегии противника от стратегий, которые позволили бы противнику тривиально выиграть, например, сделать испытательный бит зависящим от шифротекста, возможно очень сложным образом. Также такие стратегии противника не кажутся слишком полезными.

Игра продолжается следующим образом. Эксперимент выбирает ключ k. Противник поочерёдно отправляет l_c алгоритмов выборки $X_1, X_2, \ldots, X_{l_c}$. Для алгоритма выборки i эксперимент выбирает

$$(ad_i, m_i, \sigma_i, b_i) \xleftarrow{r} X_i(\sigma_{i-1}),$$

где $\sigma_0 = \perp$, затем шифрует $c_i \leftarrow E(k, ad_i, m_i)$ и отправляет c_i противнику. В конце противник выводит предположение b'. Считается, что противник выигрывает, если $b' = b_1$.

Атака с выбранным открытым текстом. Когда противнику предоставляется доступ к нескольким испытательным шифротекстам, обсуждение можно немного упростить, разрешив противнику получать шифротексты открытых текстов, выбранных противником, в дополнение к

испытательным шифротекстам. Противник, очевидно, мог бы предоставить алгоритм выборки, который всегда выводит одно сообщение, но возможность шифровать выбранные открытые тексты упрощает некоторые рассуждения. Подчёркивается, что это не увеличение возможностей противника. Поэтому атака, определённая до сих пор, называется атакой с выбранным открытым текстом (CPA).

Иногда интересно различать ситуации, когда противнику разрешено видеть шифрования выбранных открытых текстов только до получения первого испытательного шифротекста, или только после получения последнего испытательного шифротекста. Такие варианты атаки с выбранным открытым текстом далее обсуждаться не будут.

Иногда необходимо рассматривать детерминированные схемы шифрования (где шифротекст полностью определяется ключом и сообщением). В этом случае нужно предотвратить возможность получения противником одного и того же сообщения зашифрованным дважды, поскольку это обычно чрезмерно упрощает работу противника.

Атака с выбранным шифротекстом. Иногда противнику удаётся убедить одного из честных пользователей принять созданный противником шифротекст как корректный и попытаться его расшифровать. Если расшифрование успешно, противник может узнать что-то о расшифровании.

Если противнику разрешено получать расшифрование испытательного шифротекста и узнавать что-то о расшифровании, противник может узнать испытательный бит. Это делает работу противника слишком лёгкой. Это решается самым простым образом — путём отказа расшифровывать какие-либо испытательные шифротексты.

Расширенная игра работает следующим образом: эксперимент отслеживает испытательные шифротексты. Противник может отправлять шифротексты эксперименту. Если шифротекст не является испытательным, эксперимент запускает алгоритм расшифрования на шифротексте с ключом эксперимента и отправляет результат противнику. Это называется атакой с выбранным шифротекстом (CCA).

В этой игре требуется быть осторожным с ассоциированными данными, но идея в том, что успешное расшифрование шифротекста в неверном контексте — плохо и нетривиально. Следовательно, испытательные шифротексты отклоняются только тогда, когда они передаются вместе с теми ассоциированными данными, с которыми они были зашифрованы.

Как и для атаки с выбранным открытым текстом, существуют варианты атаки с выбранным шифротекстом, где противнику не разрешается запрашивать расшифрования до получения первого испытательного шифротекста или после получения последнего испытательного шифротекста. Можно также рассматривать неадаптивные варианты атаки, где противник должен отправить все шифротексты одновременно. Эти и другие варианты атаки с выбранным открытым текстом обсуждаться далее не будут.

Эксперимент для семантической безопасности выполняется следующим образом:

1. Пусть $\sigma = b = \perp$, и пусть $C = \varnothing$.
2. Выбрать ключ $k \xleftarrow{r} K$.
3. Когда противник посылает запрос (ad, X) *(испытательный)*, выполнить:
   a) Выбрать $(m, \sigma', b'') \xleftarrow{r} X(\sigma)$. $\sigma \leftarrow \sigma'$. Если $b = \perp$, то $b \leftarrow b''$.
   b) $c \leftarrow E(k, ad, m)$. $C \leftarrow C \cup \{(ad, c)\}$.
   c) Отправить c противнику.
4. Когда противник посылает запрос (ad, m) *(выбранный открытый текст)*, выполнить:
   a) $c \leftarrow E(k, ad, m)$. Отправить c противнику.
5. Когда противник посылает запрос (ad, c) *(выбранный шифротекст)*, выполнить:
   a) Если $(ad, c) \in C$, отправить $\perp$ противнику.
   b) Иначе $m \leftarrow D(k, ad, c)$ и отправить m противнику.

В конце противник выводит $b' \in \{0, 1\}$. Если в этот момент $b = \perp$, эксперимент выбирает $b \xleftarrow{r} \{0, 1\}$.

Листинг 1 – Эксперимент $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}(A)$ для игры семантической безопасности симметричной криптосистемы $\Sigma = (K, P, F, C, E, D)$ с противником A.

Семантическая безопасность. Теперь может быть определена конфиденциальность для симметричной криптосистемы, которая определяется с помощью игры между экспериментом и противником.

D

Определение 1.2

(τ, l_c, l_e, l_d)-противник против семантической безопасности для симметричной криптосистемы Σ — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 1, делая не более l_c испытательных запросов (где длина сообщений, выбираемых из заданных противником алгоритмов выборки сообщений, не зависит от состояния эксперимента), l_e запросов с выбранным открытым текстом и l_d запросов с выбранным шифротекстом, и где время работы противника и эксперимента не превышает τ.

Преимущество этого противника определяется как

$$\mathsf{Adv}^{\mathsf{sem}}_{\Sigma}(A) = 2\left|\Pr[E] - \frac{1}{2}\right|,$$

где E — событие, что b', выводимый A, равен b эксперимента.

Противник является противником с выбранным открытым текстом, если он не делает запросов с выбранным шифротекстом. В противном случае это противник с выбранным шифротекстом. В этих случаях преимущество может обозначаться как $\mathsf{Adv}^{\mathsf{sem\text{-}cpa}}_{\Sigma}(A)$ и $\mathsf{Adv}^{\mathsf{sem\text{-}cca}}_{\Sigma}(A)$.

Замечание. Слово «преимущество» используется, чтобы указать, что наш противник имеет преимущество по сравнению с противником, который просто угадывает. Некоторые авторы предпочитают различать преимущество и вероятность успеха, используя последнюю, когда противник не измеряется относительно чего-то ещё. Здесь этого делаться не будет.

Замечание. Реальные противники не заинтересованы в атаке на криптографию. Они хотят атаковать систему, которая использует криптографию. Один из способов атаковать такую сложную систему — атаковать криптографию. Как отличить атаки на систему через криптографию от атак, которые лишь случайно связаны с криптографией?

Идея состоит в том, чтобы моделировать систему и противника. При моделировании идентифицируется использование криптосистемы, которое моделируется как взаимодействия с экспериментом. Остальная часть модели системы затем объединяется с противником для создания единого противника против криптосистемы.

Если этот противник против криптосистемы имеет существенное преимущество, то это криптографическая атака. В противном случае противник против системы не атаковал криптосистему. Примеры более крупных систем будут рассмотрены позже.

Замечание. Другой способ моделировать игру — дать противнику доступ к некоторым оракулам, вместо того чтобы позволять ему разговаривать с экспериментом. В нашем случае противник получил бы доступ к оракулу шифрования, оракулу расшифрования и испытательному оракулу. Настройка эксперимента в этом случае отвечала бы за создание ключей и общих значений. Различные оракулы должны были бы взаимодействовать друг с другом. Противник посылал бы сообщения оракулам, а оракулы отвечали бы в соответствии со своей программой.

Определение игр таким образом по существу эквивалентно тому, что уже было сделано, и реальной разницы в выразительной силе нет. Какой стиль выбрать, во многом является вопросом вкуса и темперамента.

Обычно удобно определить ещё немного формального аппарата до доказательства безопасности схем, что и будет сделано в Разделах 7.2.1 и 7.2.2. Однако существует одна простая схема, безопасность которой может быть доказана без введения дополнительного аппарата.

E

Упражнение 1.1

Показать, что для любого $(\tau, 1, 0, 0)$-противника $\mathcal{A}$ против одноразового блокнота из Раздела 1.2.8, $\mathrm{Adv}^{\text{sem-cpa}}_{\text{otp}}(\mathcal{A}) = 0$ независимо от τ.

Упражнение 1.2

В этом упражнении работа противника будет усложнена. Иногда интересно рассматривать безопасность шифрования случайных сообщений. Частичная утечка сообщения в этих приложениях часто не является проблемой, поэтому целью противника является восстановление всего сообщения. Определить понятие однонаправленной безопасности для некоторого подмножества открытых текстов, которое захватывает эту идею.

Упражнение 1.3

Это упражнение продолжает обсуждение детерминированного шифрования. Пусть Σ — детерминированная симметричная криптосистема. Привести $(\tau, 1, 1, 0)$-противника против семантической безопасности с преимуществом 1 и тривиальной оценкой τ.

Эксперимент left-or-right идентичен эксперименту на Листинге 1, за исключением того, что два шага изменены следующим образом:

1. $b \xleftarrow{r} \{0,1\}$. $C \leftarrow \varnothing$.
2. Когда противник посылает запрос (ad, m_0, m_1) *(испытательный)*, выполнить:
   a) $c \leftarrow E(k, ad, m_b)$.
   b) $C \leftarrow C \cup \{(ad, c)\}$.
   c) Отправить c противнику.

Листинг 2 – Эксперимент $\mathsf{Exp}^{\mathsf{ind}}_{\Sigma}(A)$ для игры лево-правой безопасности (left-or-right) для симметричной криптосистемы $\Sigma = (K, P, F, C, E, D)$ с противником A, основанный на эксперименте с Листинга 1.

Неотличимость, или left-or-right безопасность. Определение семантической безопасности несколько сложно использовать. Теперь будет определено понятие, которое проще использовать, но которое не выглядит немедленно как “правильное” понятие безопасности для симметричной криптосистемы.

Идея в этой игре состоит в том, что испытательные запросы противника будут парами сообщений одинаковой длины. Эксперимент будет либо всегда шифровать левое сообщение, либо всегда шифровать правое сообщение. В остальном эксперимент идентичен эксперименту игры семантической безопасности.

D

Определение 1.3

(τ, l_c, l_e, l_d)-противник против left-or-right безопасности (или неотличимости) для симметричной криптосистемы Σ — это интерактивный алгоритм A, который взаимодействует с экспериментом на Рисунке 7.2, делая не более l_c испытательных запросов (пар сообщений одинаковой длины), l_e запросов выбранного открытого текста и l_d запросов выбранного шифротекста, и при этом время работы противника и эксперимента не превышает τ.

Преимущество этого противника определяется как

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(A) = 2\left|\Pr[E] - \tfrac{1}{2}\right|,$$

где E — событие, что b', выводимое A, равно b эксперимента.

Противник является противником выбранного открытого текста, если он не делает запросов выбранного шифротекста. В противном случае он является противником выбранного шифротекста. В этих случаях преимущество может обозначаться

$$\mathrm{Adv}^{\text{ind-cpa}}_{\Sigma}(A) \quad \text{и} \quad \mathrm{Adv}^{\text{ind-cca}}_{\Sigma}(A).$$

Замечание. Некоторые авторы используют слово «неотличимость» только когда количество испытательных запросов равно 1, а в остальных случаях используют название left-or-right безопасность. Мы будем считать неотличимость и left-or-right синонимами.

Мы начинаем с повторения Упражнения 1.1. Сравните сложность двух доказательств.

E

Упражнение 1.4

Показать, что для любого $(\tau, 1, 0, 0)$-противника A против одноразового блокнота из Раздела 1.2.8,

$$\mathrm{Adv}^{\text{ind-cpa}}_{\mathrm{OTP}}(A) = \tfrac{1}{2},$$

независимо от τ.

Теперь будет продолжено рассмотрение взаимосвязи между семантической безопасностью и неотличимостью, доказывая, что любой противник неотличимости может быть превращён в столь же хорошего противника против семантической безопасности. Интуитивно, семантическая безопасность влечёт неотличимость. Заметим, что мы хотим получить нечто, что проще использовать, чем семантическая безопасность, но всё же обеспечивает ту же безопасность. Следовательно, это не то направление импликации, которое нам нужно! Желаемая импликация появится позже.

Идея доказательства заключается в том, что left-or-right испытательный запрос, состоящий из пары сообщений одинаковой длины, может быть преобразован в алгоритм выборки, который либо выбирает левое сообщение, либо правое. С небольшим вниманием можно убедиться, что последовательность создаваемых алгоритмов выборки всегда шифрует левое или всегда шифрует правое. С таким дополнением эксперимент семантической безопасности будет создавать шифротекст с точно тем же распределением, что и эксперимент неотличимости создавал бы в ответ на исходный запрос.

Техническая реализация этой идеи принимает форму редукции, в том смысле, что будет сведена задача атаки семантической безопасности к задаче атаки неотличимости. Между противником неотличимости и экспериментом семантической безопасности вставляется промежуточное звено, чья задача — интерпретировать запросы A и ответы эксперимента. Это промежуточное звено делает эксперимент семантической безопасности $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$ похожим для A на эксперимент неотличимости $\mathrm{Exp}^{\mathrm{ind}}_{\Sigma}$, в то время как делает A похожим на противника семантической безопасности для эксперимента $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$.

Это промежуточное звено можно в некотором смысле назвать редукцией. Однако это очень простая форма редукции, и поскольку задача промежуточного звена — симулировать другие типы участников, мы будем называть его *симулятором*.

Утверждение 1.1

Пусть $\mathcal{A}$ является (τ, l_c, l_e, l_d)-противником против неотличимости для симметричной криптосистемы Σ. Тогда $\mathcal{B}$, заданная на рисунке 3 и листинге 3, является (τ', l_c, l_e, l_d)-противником против семантической безопасности для Σ, где τ' по существу равна τ, и

$$\mathrm{Adv}^{\mathrm{sem}}_{\Sigma}(\mathcal{B}) \;=\; \mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(\mathcal{A}).$$

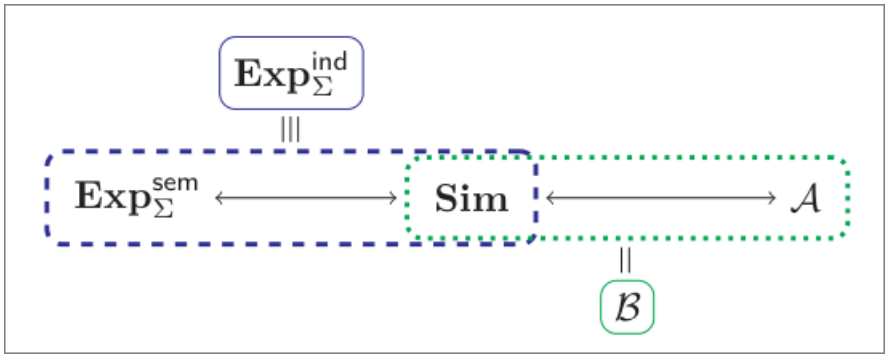

Рисунок 1 – Идея, используемая в доказательстве Утверждения 1.1. Часть внутри штриховой линии ведёт себя как эксперимент неотличимости, что означает, что противник должен вести себя ожидаемым образом. Часть внутри пунктирной линии становится противником против семантической безопасности.

Противник B запускает копию A и следующий симулятор Sim:

- Когда A посылает испытательный запрос (ad, m_0, m_1), симулятор создаёт X, который:
  - при входе $\perp$ выбирает $b \xleftarrow{r} \{0,1\}$ и выводит (m_b, b, b); и
  - при входе $b \in \{0,1\}$ выводит (m_b, b, b).

  Симулятор посылает (ad, X) в эксперимент семантической безопасности. Когда эксперимент отвечает c, симулятор отправляет c алгоритму A.
- Симулятор перенаправляет остальные запросы в эксперимент и пересылает ответы алгоритму A.

Когда A выводит b', алгоритм B выводит b'.

Листинг 3 – Симулятор и противник, используемые в доказательстве Утверждения 7.1.

Доказательство. Сначала доказывается, что комбинация эксперимента семантической безопасности $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$ и симулятора Sim с Листинга 3 ведёт себя в точности так же, как эксперимент неотличимости. Затем доказывается, что корректная догадка совпадает в обоих случаях. Это устанавливает, что преимущества идентичны.

Поскольку Sim просто перенаправляет запросы выбранного открытого текста и выбранного шифротекста, такие запросы обрабатываются $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$ и Sim тем же образом, каким они обрабатываются в $\mathrm{Exp}^{\mathrm{ind}}_{\Sigma}$. Непосредственным просмотром видно, что Sim и $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$ всегда шифруют m_b при обработке испытательного запроса, точно так же, как это делает эксперимент неотличимости. Следовательно, испытательные запросы обрабатываются одинаковым образом, при фиксированном b.

Эксперимент неотличимости выбирает испытательный бит в начале. Эксперимент семантической безопасности выбирает испытательный бит при обработке первого испытательного запроса или в конце игры. Однако до первого испытательного запроса ничего не зависит от испытательного бита, и в обоих случаях бит выбирается из равномерного распределения на $\{0, 1\}$.

Непосредственным просмотром видно, что догадка A корректна тогда и только тогда, когда корректна догадка B. Следовательно, преимущество B при взаимодействии с $\mathrm{Exp}^{\mathrm{sem}}_{\Sigma}$ равно преимуществу A при взаимодействии с $\mathrm{Exp}^{\mathrm{ind}}_{\Sigma}$.

По построению, оба противника делают одинаковое количество испытательных запросов, запросов выбранного открытого текста и запросов выбранного шифротекста. Что касается времени работы, имеется небольшая добавка на каждый запрос, поскольку должна быть выполнена пересылка через Sim, но для любого нетривиального противника такой накладные расходы можно безопасно игнорировать. Следовательно, τ' по существу равна τ. □

Утверждение о времени выполнения в приведённом выше утверждении, выраженное словами «по существу равны», является примечательно неточным. Эту формулировку можно сделать точной, если выбрать некоторую базовую вычислительную модель. При некоторых оптимизациях накладные расходы, вероятно, будут пренебрежимо малы. Однако требуемая детализация будет значительной.

В качестве альтернативы можно выбрать приближённое решение и утверждать, что стоимость линейна по числу запросов, обычно обозначаемая $O(l_c + l_e + l_d)$. Но для любой разумной вычислительной модели эта стоимость будет линейна не только по числу запросов, но также по длине отдельных сообщений, так что в указанное приближение пришлось бы включить суммарную длину сообщений. Таким образом, стоимость приблизительно линейна по числу запросов плюс общий объём данных, зашифрованных или расшифрованных.

В общем случае приходится предполагать, что противник может выделить на атаку системы значительно больше ресурсов, чем пользователи системы готовы выделить на её работу. Из этого следует, что ограничение на время выполнения будет значительно больше общего объёма зашифрованных и расшифрованных данных. Отсюда следует, что разница между двумя временами выполнения будет относительно мала.

Существуют редкие случаи, когда эта разница не может быть проигнорирована, например если по каким-либо причинам рассматриваются только противники с очень небольшим временем работы. Вывод состоит в том, что как компромисс между точностью и простотой используется выражение «по существу равны», чтобы указать, что эти две величины не равны, но разница между ними относительно мала и обычно может быть проигнорирована. В каждом доказательстве необходимо аккуратно проверять, какова эта разница, действительно ли она относительно мала и действительно ли она обычно может быть проигнорирована.

Замечание. Ещё один важный момент относительно времени выполнения, упомянутого в выше приведённом утверждении, заключается в том, что рассматривается глобальное время выполнения, то есть время работы и противника, и эксперимента, а не только время работы противника. Как обсуждалось в предыдущем замечании, эксперимент обычно моделирует честных пользователей, чьё суммарное время работы будет значительно меньше времени работы противника, и поэтому этот вклад мог бы быть проигнорирован. Однако далее возникнут технические проблемы доказательств, в которых локальная оценка времени вызвала бы существенные трудности учёта. Использование глобального времени выполнения упрощает ситуацию.

E

Упражнение 1.5

Это упражнение продолжает теорию детерминированного шифрования и результат из Упражнения 1.3. Определить вариант неотличимости, который мог бы выполняться для детерминированного шифрования.

Безопасность real-or-random. Вводится понятие, которое, возможно, выглядит ещё более далёким от «корректного» понятия безопасности для симметричной криптосистемы, однако оно окажется существенно проще для использования в дальнейшем.

Эксперимент real-or-random идентичен эксперименту на листинге 1, за исключением того, что два шага изменяются следующим образом:

1. $b \xleftarrow{r} \{0,1\}$. $C \leftarrow \varnothing$.
2. Когда противник посылает запрос (ad, m_0) *(испытательный)*, выполнить:
   a) Выбрать $m_1 \xleftarrow{r} \{\, m \in P \mid m$ имеет ту же длину, что $m_0 \,\}$.
   b) $c \leftarrow E(k, ad, m_b)$.
   c) $C \leftarrow C \cup \{(ad, c)\}$.
   d) Отправить c противнику.

Листинг 4 – Эксперимент для игры real-or-random безопасности для симметричной криптосистемы $\Sigma = (K, P, F, C, E, D)$, основанный на эксперименте с Листинга 1.

Идея игры состоит в том, что испытательные запросы противника содержат одиночные сообщения. Эксперимент либо всегда шифрует выбранное противником сообщение, либо всегда шифрует случайно выбранное сообщение той же длины. Помимо изменений в учёте запросов и обработке испытательного запроса, эксперимент идентичен экспериментам семантической безопасности и неотличимости.

Вопреки первоначальному замечанию, это в определённом смысле вполне разумное понятие безопасности. Если противник не способен определить, содержит ли шифротекст конкретное сообщение или случайные данные, то он вряд ли может получить какую-либо полезную информацию из шифротекста. Этот подход будет играть важную роль.

D

Определение 1.4: Противник против безопасности real-or-random

(τ, l_c, l_e, l_d)-противник против безопасности real-or-random для симметричной криптосистемы Σ — это интерактивный алгоритм A, взаимодействующий с экспериментом на Рисунке 7.5, делающий не более l_c испытательных запросов, l_e запросов выбранного открытого текста и l_d запросов выбранного шифротекста, причём время выполнения противника и эксперимента не превосходит τ. Преимущество определяется как

$$\mathrm{Adv}^{\mathrm{ror}}_{\Sigma}(A) = 2\left|\Pr[E] - \tfrac{1}{2}\right|,$$

где E — событие, состоящее в совпадении b' (выхода A) с битом b эксперимента. Противник является CPA-противником, если не делает запросов выбранного шифротекста; иначе он является CCA-противником. Соответствующие обозначения преимуществ: $\mathrm{Adv}^{\text{ror-cpa}}_{\Sigma}(A)$, $\mathrm{Adv}^{\text{ror-cca}}_{\Sigma}(A)$.

Замечание. Противник не может получить преимущество 1 в игре real-or-random, поскольку может случиться, что случайное сообщение совпадает с сообщением, выбранным противником. Это событие маловероятно, однако оно объясняет, почему некоторые получаемые позднее оценки строго меньше единицы.

Доказывается, что любой противник против real-or-random может быть преобразован в столь же эффективного противника против неотличимости. Следовательно, неотличимость имплицирует безопасность real-or-random.

Идея доказательства аналогична идее доказательства Утверждения 7.1: испытательный запрос real-or-random, содержащий одно сообщение, может быть преобразован в left-or-right запрос путём выбора случайного сообщения подходящей длины. После этого эксперимент left-or-right создаёт шифротекст с той же распределённостью, что и эксперимент real-or-random.

P

Утверждение 1.2

Пусть A является (τ, l_c, l_e, l_d)-противником против безопасности real-or-random для симметричной криптосистемы Σ. Тогда существует (τ', l_c, l_e, l_d)-противник B против неотличимости для Σ, где τ' по существу равна τ, и

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(B) = \mathrm{Adv}^{\mathrm{ror}}_{\Sigma}(A).$$

E

Упражнение 1.6

Доказать Утверждение 1.2.

Следующий результат показывает, что семантическая безопасность, неотличимость и безопасность real-or-random являются по существу эквивалентными понятиями безопасности. Это означает, что можно выбирать то понятие, которое наиболее удобно при анализе симметричных криптосистем.

Этот результат демонстрирует практическую ценность безопасности real-or-random: замена шифрований содержательных сообщений на шифрования случайных сообщений является чрезвычайно мощной техникой.

Удобно сначала доказать небольшую лемму, связывающую преимущество противника с разницей в его поведении в двух различных условиях (измеряемой вероятностью того, что противник выводит 1). Благодаря этому противник может быть помещён в две тесно связанные ситуации, и сразу получаются утверждения о возможной разнице в поведении.

L

Лемма 1.1

Пусть A и Exp являются интерактивными алгоритмами. Пусть Exp выбирает бит $b \xleftarrow{r} \{0,1\}$, затем Exp и A взаимодействуют, после чего A выводит бит $b' \in \{0,1\}$. Обозначим через Exp_i вариант Exp, который всегда выбирает $b = i$. Пусть E — событие, состоящее в том, что $b = b'$ после взаимодействия, и пусть F_i — событие, состоящее в том, что $b' = 1$ после взаимодействия между Exp^i и A. Тогда выполняется равенство

$$2\big|\Pr[E] - \tfrac{1}{2}\big| \;=\; \big|\Pr[F_0] - \Pr[F_1]\big|.$$

Доказательство. Пусть E_i — событие, состоящее в том, что $b = b'$ после взаимодействия между Exp^i и A. Вычисляется:

$$\Pr[E] = \Pr[E_0]\Pr[b=0] + \Pr[E_1]\Pr[b=1] = \tfrac{1}{2}\big((1 - \Pr[F_0]) + \Pr[F_1]\big)$$

$$= \tfrac{1}{2} - \tfrac{1}{2}\big(\Pr[F_1] - \Pr[F_0]\big),$$

откуда заключение немедленно следует. □

Утверждение 1.3

Пусть A является (τ, l_c, l_e, l_d)-противником против семантической безопасности для симметричной криптосистемы Σ. Тогда B, заданный на Листинге 5, является (τ', l_c, l_e, l_d)-противником против безопасности real-or-random для Σ, где τ' по существу равна τ, и

$$\mathrm{Adv}^{\mathrm{ror}}_{\Sigma}(B) = \tfrac{1}{2}\,\mathrm{Adv}^{\mathrm{sem}}_{\Sigma}(A).$$

Противник B запускает копию A и следующий симулятор Sim:

- В начале симулятор устанавливает $\sigma = \bar{b} = \perp$.
- Когда A посылает испытательный запрос (ad, X), симулятор выбирает $(m_0, \sigma', b'') \xleftarrow{r} X(\sigma)$. Если $\bar{b} = \perp$, алгоритм B устанавливает $\bar{b} = b''$. Затем симулятор посылает испытательный запрос (ad, m_0) в эксперимент real-or-random. Когда эксперимент отвечает шифротекстом c, симулятор пересылает c алгоритму A.
- Симулятор перенаправляет все остальные запросы в эксперимент и пересылает ответы алгоритму A.

Если алгоритм A выводит $\bar{b}'$, алгоритм B выводит $b' = 1$, если $\bar{b}' = \bar{b}$, и выводит $b' = 0$ в противном случае. Если A превышает свои пределы, то B выбирает $b' \xleftarrow{r} \{0, 1\}$ и выводит b'.

Листинг 5 – Симулятор и противник, используемые в доказательстве Утверждения 1.3.

Доказательство. Поскольку A может работать не более времени τ, максимальное время работы B и эксперимента по существу совпадает с τ для любого нетривиального противника.

Требуется вычислить вероятность $\Pr[b = b']$, когда B взаимодействует с экспериментом real-or-random, и Лемма 1.1 даёт

$$2\left|\Pr[b = b'] - \tfrac{1}{2}\right| = \left|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\right|.$$

Если $b = 0$ (что означает, что эксперимент real-or-random всегда будет шифровать заданное испытательное сообщение), то поскольку Sim выбирает испытательное сообщение точно так же, как эксперимент семантической безопасности выбирает сообщение для шифрования, A не превышает своих

пределов, и выполняется

$$2\big|\Pr[\bar{b}' = \bar{b} \mid b = 0] - \tfrac{1}{2}\big| = \mathrm{Adv}^{\mathrm{sem}}_{\Sigma}(A).$$

Кроме того,

$$\Pr[\bar{b}' = \bar{b} \mid b = 0] = \Pr[b' = 1 \mid b = 0].$$

С другой стороны, если $b = 1$ (что означает, что эксперимент real-or-random всегда будет шифровать сообщения, независимые от испытательного сообщения), то поскольку $\bar{b}$ участвует только в выборе испытательных сообщений, $\bar{b}$ является независимым от всего, что наблюдает A. Следовательно,

$$\Pr[\bar{b}' = \bar{b} \mid b = 1] = \tfrac{1}{2}.$$

Иными словами, поскольку A не имеет никакой информации о бите $\bar{b}$, выбранном B при $b = 1$, получаем

$$\Pr[\bar{b}' = \bar{b} \mid b = 1] = \Pr[b' = 1 \mid b = 1] = \tfrac{1}{2}.$$

(Это также выполняется, если A превышает свои пределы.)

Объединяя указанное, получаем

$$\mathrm{Adv}^{\mathrm{ror}}_{\Sigma}(B) = \big|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\big|$$

$$= \tfrac{1}{2} \cdot 2\big|\Pr[\bar{b}' = \bar{b} \mid b = 0] - \tfrac{1}{2}\big| = \tfrac{1}{2}\,\mathrm{Adv}^{\mathrm{sem}}_{\Sigma}(A),$$

чем утверждение доказывается. □

Замечание. Следует ещё раз отметить, что множитель $1/2$ является существенным, поскольку в эксперименте real-or-random случайный вариант может зашифровать то же самое сообщение, которое было бы зашифровано в реальном варианте. Отсюда следует, что противник real-or-random не может достичь преимущества, равного 1.

Замечание. Это доказательство показывает необходимость осторожности при работе со временем выполнения при построении противников. Причина состоит в том, что алгоритмы, рассматриваемые в данном контексте, гарантированно завершают работу в пределах заданного ограничителя времени только в том случае, если они получают ожидаемый

ввод. Если алгоритму передаётся ввод, которого он не ожидает (например, шифрования равномерно случайных сообщений вместо шифрований сообщений, выбираемых из определённого распределения), нельзя ожидать, что алгоритм сохранит обещанное время выполнения или другие ограничения. Если бы не была добавлена проверка пределов, противник мог бы не завершить работу при $b = 1$, и тогда противник вообще не был бы корректно определён.

Утверждения 1.1, 1.2 и 1.3 означают, что три введённых понятия безопасности по существу эквивалентны. Множитель $1/2$ в этой связи в основном несуществен.

1.2 Одного испытательного запроса достаточно, возможно

В длительном обсуждении в Разделе 1.1 безопасность изначально определялась в терминах одного испытательного запроса, после чего были разрешены несколько испытательных запросов. Теперь показывается, что в определённом смысле достаточно доказывать безопасность для одного испытательного запроса. Однако сразу следует сделать предупреждение: доказательство безопасности для одного испытательного запроса даёт более слабый общий результат. При прочих равных условиях доказательство для нескольких испытательных запросов лучше, чем доказательство для одного.

Эта теорема также наглядно иллюстрирует важную доказательную технику, а именно гибридный аргумент. Идея состоит в построении последовательности очень похожих игр, после чего разница между соседними играми увязывается с одним свойством, а разница между крайними элементами последовательности — с другим.

Утверждение 1.4

Пусть A является (τ, l_c, l_e, l_d)-противником против неотличимости для Σ. Тогда B, заданный на Листинге 6, является $(\tau', 1, l_e + l_c - 1, l_d)$-противником против неотличимости для Σ, где τ' по существу равна τ, и

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(A) \;\leq\; l_c \,\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(B).$$

Противник B запускает копию A и следующий симулятор Sim:

– В начале симулятор выбирает $j \xleftarrow{r} \{1, 2, \ldots, l_c\}$.
– Когда A делает свой i-й испытательный запрос (ad, m_0, m_1):
– если $i < j$, симулятор посылает (ad, m_1) как запрос на выбранный открытый текст;
– если $i > j$, симулятор посылает (ad, m_0) как запрос на выбранный открытый текст;
– если $i = j$, симулятор посылает (ad, m_0, m_1) как испытательный запрос. После того как эксперимент отвечает шифротекстом c, симулятор сохраняет пару (ad, c) и пересылает шифротекст алгоритму A.
– Если A делает запрос на выбранный шифротекст (ad, c) и такая пара была сохранена ранее, симулятор немедленно отвечает $\perp$. Иначе симулятор перенаправляет запрос своему эксперименту и пересылает ответ алгоритму A.
– Симулятор перенаправляет все остальные запросы в эксперимент и пересылает ответы алгоритму A.

Если алгоритм A выводит b', алгоритм B выводит то же значение b'. Если A превышает свои пределы, то алгоритм B выбирает $b' \xleftarrow{r} \{0, 1\}$ и выводит b'.

Листинг 6 – Симулятор и $(\tau', 1, l_e + l_c - 1, l_d)$-противник B против неотличимости, построенные на основе (τ, l_c, l_e, l_d)-противника A против неотличимости.

Доказательство. Поскольку алгоритм A может работать не более времени τ, максимальное время работы B и игры по существу совпадает с τ для любого нетривиального противника.

Обозначим через F_{j_0} событие, состоящее в том, что симулятор выбирает значение j_0 для j. Обозначим через E_{j_0,b_0} событие, состоящее в том, что эксперимент неотличимости выбирает b_0 для b, симулятор выбирает значение j_0 для j, после чего B выводит $b' = 1$.

Сначала заметим: если $b = 0$ и симулятор выбирает значение 1 для j, то алгоритм A получит шифрование левого сообщения для всех своих испытательных запросов. Аналогично, если $b = 1$ и симулятор выбирает значение l_c для j, то A получит шифрование правого сообщения для всех своих испытательных запросов. В этих случаях известно, что время работы

эксперимента неотличимости и противника не превышает τ и что A никогда не выходит за пределы. Это означает, что алгоритм B никогда не останавливается со случайным выводом. Следовательно,

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(A) = \big|\Pr[E_{1,0}] - \Pr[E_{l_c,1}]\big|. \tag{1.1}$$

Если $b = 1$ и симулятор выбирает значение j_0 для j, то A будет получать шифрования правого сообщения для первых j_0 испытательных запросов, и шифрования левого сообщения для оставшихся. Однако если $b = 0$ и симулятор выбирает значение $j_0 + 1$ для j, то снова A будет получать шифрования правого сообщения для первых j_0 испытательных запросов, и шифрования левого сообщения для оставшихся. Поскольку других возможных различий между этими двумя случаями нет, выполняется

$$\Pr[E_{j_0,1}] = \Pr[E_{j_0+1,0}].$$

Наконец, заметим, что

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(B) = \frac{1}{l_c}\sum_{j=1}^{l_c}\big|\Pr[E_{j,0}] - \Pr[E_{j,1}]\big|.$$

Теперь, используя (1.1) и телескопическую сумму, получаем

$$\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(A) = \big|\Pr[E_{1,0}] - \Pr[E_{1,1}] + \Pr[E_{2,0}] - \cdots - \Pr[E_{l_c,1}]\big|$$

$$\leq \sum_{j=1}^{l_c}\big|\Pr[E_{j,0}] - \Pr[E_{j,1}]\big| = l_c\,\mathrm{Adv}^{\mathrm{ind}}_{\Sigma}(B),$$

чем утверждение завершается. □

Замечание. Фактор потери преимущества l_c вероятно является существенным для общего утверждения. Однако это не накладывает ограничений на то, что может быть доказано с использованием нескольких испытательных запросов, и следует стремиться доказывать утверждения без этой потери.

Гибридное доказательство использует одного противника. Можно было бы использовать l_c однократных противников, параметризованных значением j. Тогда следовало бы доказать, что их среднее преимущество должно быть не меньше $1/l_c$ от преимущества многократного противника. Из этого следовало бы, что по крайней мере один из противников имеет преимущество, большее или равное среднему. Выбор структуры доказательства является в определённой степени вопросом стиля и предпочтений, но иногда один подход оказывается проще другого.

Доказательство для безопасности вида real-or-random по существу совпадает с доказательством для неотличимости. Однако прямое доказательство для семантической безопасности было бы несколько более сложным, поскольку нельзя легко контролировать испытательный бит. Вместо этого, чтобы получить соответствующий результат для семантической безопасности, берётся многократный противник против семантической безопасности, затем с помощью Утверждений 1.3 и 1.2 строится противник против неотличимости с половиной преимущества. После этого с помощью Утверждения 1.4 получается однократный противник против неотличимости с меньшим преимуществом. Наконец, с помощью Утверждения 1.1 строится однократный противник против семантической безопасности. Это иллюстрирует силу теорем, устанавливающих редукции между различными понятиями безопасности.

1.3 Шифротексты, похожие на случайный шум

Некоторые криптосистемы фактически обеспечивают более сильное понятие real-or-random, чем определённое выше. Идея заключается в том, что шифрования выбранных сообщений трудно отличить не только от шифрований случайных сообщений, но и от случайности, полностью независимой не только от выбранного сообщения и ассоциированных данных, но и от самого секретного ключа.

Это свойство довольно удобно, когда симметричная криптография используется как часть более крупной системы, но также имеет и прямые применения. Одно из применений — скрывать шифротекст в случайном шуме. Другое — показывать, что шифротексты псевдослучайны, что позволяет использовать результаты, требующие случайности (такие как лемма о остаточном хэшировании).

Эксперимент для игры R-rnd с противником A идентичен эксперименту с Листинга 1, за исключением того, что два шага изменены следующим образом:

а) $b \xleftarrow{r} \{0,1\}$. $C \leftarrow \varnothing$.

б) Когда противник посылает запрос (ad, m) (испытательный), выполняется:

(a) Если $b = 0$, зашифровать $c \leftarrow E(k, ad, m)$.

(b) Если $b = 1$, выбрать $c \xleftarrow{r} R_\ell$, где ℓ — длина m.

(c) $C \leftarrow C \cup \{(ad, c)\}$.

(d) Послать c противнику.

Листинг 7 – Эксперимент для игры R-random безопасности для симметричной криптосистемы $\Sigma = (K, P, F, C, E, D)$ и семейства шума R на C, основанный на эксперименте с Листинга 1.

D

Определение 1.5

Пусть $\Sigma = (K, P, F, C, E, D)$ — симметричная криптосистема. Семейство шума R — это семейство алгоритмов выборки, индексированное неотрицательными целыми числами.

(τ, l_c, l_e, l_d)-противник против R-random для симметричной криптосистемы Σ — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 7, делая не более l_c испытательных запросов, l_e запросов на выбранный открытый текст и l_d запросов на выбранный шифротекст, причём время работы противника и эксперимента не превосходит τ.

Преимущество этого противника определяется как

$$\mathrm{Adv}_{\Sigma}^{R\text{-rnd}}(A) = 2\left|\Pr[E] - 1/2\right|,$$

где E — событие, что b' выводимый A равен биту b, выбранному экспериментом.

E

Упражнение 1.7

Пусть Σ — криптосистема с множеством шифротекстов C и пусть R — семейство шума на C. Докажите, что если A — любой (τ, l_c, l_e, l_d)-противник против real-or-random безопасности, то существует (τ', l_c, l_e, l_d)-противник против R-random безопасности для криптосистемы, где τ' по существу совпадает с τ, а их преимущества практически одинаковы (с точностью до малого множителя).

Наиболее интересный случай — когда шифротексты являются битовыми строками, а семейство шума R_ℓ представляет собой равномерное распределение на множестве битовых строк некоторой длины, связанной с ℓ. Если имеется R-random безопасность, то шифротексты выглядят как случайные битовые строки, что может быть весьма полезно в доказательствах безопасности.

Следующее упражнение опровергает обратное утверждение из Упражнения 7.7. Если шифротексты выглядят случайными, то имеется защищённое шифрование. Но защищённое шифрование не влечёт случайно выглядящие шифротексты.

E

Упражнение 1.8

Пусть Σ — произвольная криптосистема, в которой шифротексты являются битовыми строками. Постройте новую криптосистему Σ', в которой шифротексты также являются битовыми строками, удовлетворяющую двум условиям: шифротексты тривиально отличимы от случайных битовых строк; и для любого противника A против real-or-random безопасности криптосистемы Σ' существует противник B против real-or-random безопасности криптосистемы Σ, имеющий то же преимущество и по существу то же время работы.

Замечание. Случайно выглядящие шифротексты, и в меньшей степени real-or-random, связаны с общим подходом к безопасности, часто называемым симулируемостью. Идея заключается в том, что должно быть возможно имитировать действия честных сторон без какого-либо знания о том, что они делают, кроме некоторой заложенной утечки. Этот подход будет необходим в Главах 11 и 12. В общем виде он позволяет получить весьма мощные теоремы о композиции, но полностью этот подход развиваться не будет.

1.4 Целостность

Для многих приложений целостность является более важной, чем конфиденциальность. Неофициально целостность имеется тогда, когда противник не может создавать допустимые шифротексты, которые дешифруются в новые сообщения. Будут обсуждены варианты этих понятий.

Будут определены два понятия целостности. Первое, целостность открытого текста, утверждает, что противник не может создать шифротекст, который дешифруется в новое сообщение, то есть сообщение, ранее не отправленное как запрос на выбранный открытый текст. Это интуитивно соответствует тому типу целостности, который требуется в приложениях.

Второе понятие целостности, целостность шифротекста, утверждает, что противник не может создать новый допустимый шифротекст, то есть шифротекст, ранее не возвращённый в ответ на запрос открытого текста. Интуитивно это кажется слишком сильным для приложений, однако с этим понятием легче работать, и оно будет использоваться в доказательствах. Также оказывается, что для многих приложений более сильное понятие безопасности является более безопасным и более удобным.

Эксперимент целостности $\mathrm{Exp}_{\Sigma}^{\mathrm{int}}$ выполняется следующим образом:

а) Выбрать ключ $k \xleftarrow{r} K$.

б) Пусть $M := \varnothing$ и $C := \varnothing$.

в) Когда противник посылает запрос (ad, m) (выбранный открытый текст):

(a) $c := E(k, ad, m)$.

(b) $M := M \cup \{(ad, m)\}, \quad C := C \cup \{(ad, c)\}$.

(c) Послать c противнику.

г) Когда противник посылает запрос (ad, c) (тест):

(a) Вычислить $m := D(k, ad, c)$ и послать m противнику.

Листинг 8 – Эксперимент для игр на целостность для симметричной криптосистемы $\Sigma = (K, P, F, C, E, D)$.

D

Определение 1.6

(τ, l_e, l_d)-противник против целостности для симметричной криптосистемы Σ — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 8, делая не более l_e запросов на выбранный открытый текст и l_d тестовых запросов, причём время работы противника и эксперимента не превосходит τ.
Преимущества по целостности открытого текста и шифротекста определяются как

$$\mathrm{Adv}^{\text{int-ptxt}}_{\Sigma}(A) = \Pr[E] \quad \text{и} \quad \mathrm{Adv}^{\text{int-ctxt}}_{\Sigma}(A) = \Pr[F],$$

где событие E состоит в том, что для некоторого тестового запроса (ad, c) дешифрование $m \neq \perp$ и $(ad, m) \notin M$, а событие F состоит в том, что для некоторого тестового запроса $(ad, c) \notin C$ дешифрование не равно $\perp$. Шифротексты в событиях E и F называются подделками.

Неофициально схема считается обладающей целостностью открытого текста, если имеется разумный аргумент того, что любой осуществимый противник по целостности не имеет значимого преимущества целостности открытого текста. Целостность шифротекста имеет соответствующее неофициальное значение. Если известны осуществимые противники с существенным преимуществом, то схема не обладает целостностью открытого/шифротекста.

Рассмотрим события E и F в приведённом выше определении. Поскольку событие E не может произойти, если не произошло событие F, очевидно, что для любого противника по целостности его преимущество по открытому тексту не меньше его преимущества по шифротексту. Теперь будет доказано, что обратное неверно, что показывает: в отличие от понятий конфиденциальности, эти два понятия целостности не эквивалентны.

Симметричная криптосистема $\Sigma = (K, P, F, C, E, D)$ имеет те же множества ключей и открытых текстов, что и Σ_0. Множество шифротекстов задаётся как $C := C_0 \times \{0, 1\}$.

Алгоритмы шифрования и расшифрования работают следующим образом:

- При вводе k, ad и m алгоритм E вычисляет $c := E_0(k, ad, m)$, выбирает $i \xleftarrow{r} \{0, 1\}$ и выводит (c, i).
- При вводе k, ad и (c, i) алгоритм D вычисляет и выводит $m := D_0(k, ad, c)$.

Листинг 9 – Симметричная криптосистема $\Sigma = (K, P, F, C, E, D)$ без целостности шифротекста, построенная на основе симметричной криптосистемы $\Sigma_0 = (K, P, F, C_0, E_0, D_0)$.

Утверждение 1.5

Пусть Σ_0 — произвольная симметричная криптосистема, и пусть Σ — криптосистема, построенная на основе Σ_0 как указано на Листинге 9. Тогда существует $(\tau, 1, 1)$-противник A против целостности для Σ, имеющий преимущество по шифротексту 1, где τ тривиально. Кроме того, для любого (τ', l_e, l_c)-противника B против целостности для Σ существует (τ'', l_e, l_c)-противник B' против целостности для Σ_0, где τ'' по существу равен τ', и такой, что

$$\mathrm{Adv}^{\text{int-ptxt}}_{\Sigma_0}(B') = \mathrm{Adv}^{\text{int-ptxt}}_{\Sigma}(B).$$

Доказательство. Противник A против целостности шифротекста для Σ строится непосредственно. Сначала он делает запрос на выбранный открытый текст для произвольного сообщения, получая в ответ (c, i). Затем он посылает тестовый запрос $(c, 1 - i)$. Очевидно, что его преимущество по целостности шифротекста равно 1, время работы пренебрежимо мало, и требуется один запрос на выбранный открытый текст и один тестовый запрос.

Теперь пусть B — противник целостности для Σ. Противник B' запускает копию B и симулятор Sim, работающий следующим образом:

- Когда B делает запрос на выбранный открытый текст (ad, m), симулятор пересылает (ad, m) в эксперимент по целостности и получает в ответ c. Затем выбирается $i \xleftarrow{r} \{0, 1\}$, и B отправляется (c, i).

– Когда B делает тестовый запрос (ad, c, i), симулятор пересылает (ad, c) в эксперимент по целостности и получает m в ответ. Затем m отправляется B.

Для каждого запроса присутствует небольшой накладной расход, так как симулятор должен пересылать данные, но для любого нетривиального противника этот расход можно безопасно игнорировать. Следовательно, τ'' по существу равен τ'.

Путём непосредственного анализа видно, что совокупная обработка запросов на выбранный открытый текст и тестовых запросов симулятором и экспериментом целостности для Σ_0 идентична обработке тех же запросов экспериментом целостности для Σ.

Отсюда следует, что если B создаёт шифротекст, который расшифровывается в новый открытый текст для Σ, то можно вывести шифротекст, который расшифровывается в новый открытый текст для Σ_0. Следовательно, преимущества по открытому тексту у B' и B совпадают. □

1.5 Конфиденциальность при выбранном открытом тексте и целостность шифротекста достаточны.

Основная теорема о целостности шифротекста является важной теоремой, поскольку она упрощает анализ схем, так как позволяет рассматривать запросы на выбранный шифротекст отдельно от испытательных запросов и запросов на выбранный открытый текст.

Сначала доказывается небольшая лемма о вероятности зависимых событий. Это мощная лемма, поскольку она позволяет ограничить расхождение двух игр, выделив одиночное исключительное событие, которое может вызвать их расхождение.

L

Лемма 1.2

Пусть E_0, E_1, F_0 и F_1 — события такие, что

$$\Pr[F_0] = \Pr[F_1] \quad \text{и} \quad \Pr[E_0 \mid \neg F_0] = \Pr[E_1 \mid \neg F_1].$$

Тогда

$$\big|\Pr[E_0] - \Pr[E_1]\big| \leq \Pr[F_0].$$

Доказательство. Вычисляется:

$$\begin{aligned}\big|\Pr[E_0] - \Pr[E_1]\big| &= \big|\Pr[E_0 \mid F_0]\Pr[F_0] + \Pr[E_0 \mid \neg F_0]\Pr[\neg F_0] \\ &\qquad - \Pr[E_1 \mid F_1]\Pr[F_1] - \Pr[E_1 \mid \neg F_1]\Pr[\neg F_1]\big| \\ &= \big|\Pr[E_0 \mid F_0]\Pr[F_0] - \Pr[E_1 \mid F_1]\Pr[F_1]\big| \\ &= \Pr[F_0] \cdot \big|\Pr[E_0 \mid F_0] - \Pr[E_1 \mid F_1]\big| \leq \Pr[F_0].\end{aligned}$$

□

Теорема 1.1

Пусть Σ — симметричная криптосистема. Пусть A — (τ, l_c, l_e, l_d)-противник против неотличимости. Тогда существуют $(\tau_1', l_c, l_e, 0)$-противник B_1 против неотличимости и $(\tau_2', l_c + l_e, l_d)$-противник B_2 против целостности, где τ_1' и τ_2' по существу равны τ, такие что

$$\mathrm{Adv}^{\text{ind-cca}}_{\Sigma}(A) \leq \mathrm{Adv}^{\text{ind-cpa}}_{\Sigma}(B_1) + 2\,\mathrm{Adv}^{\text{int-ctxt}}_{\Sigma}(B_2).$$

Доказательство. Сначала описываются противники. Противник неотличимости B_1 запускает копию A и следующий симулятор Sim_1.

– Когда A делает испытательный или запрос на выбранный открытый текст, Sim_1 перенаправляет запрос в эксперимент неотличимости. Он ведёт запись (ad, m, c) запросов на выбранный открытый текст и ответов.

– Когда A делает запрос на выбранный шифротекст (ad, c), Sim_1 проверяет, есть ли у него запись (ad, m, c) для некоторого m. Если да, Sim_1 отправляет m A. Иначе он отправляет $\perp$ A.

Если A выводит b', B_1 выводит b'. Если A превышает свои пределы, B_1 выбирает $b' \xleftarrow{r} \{0,1\}$ и выводит b'.

Противник целостности B_2 запускает копию A и следующий симулятор Sim_2.

– Sim_2 выбирает $b \xleftarrow{r} \{0,1\}$.

– Когда A посылает испытательный запрос (ad, m_0, m_1), Sim_2 посылает запрос на выбранный открытый текст (ad, m_b). Получив ответ c, он записывает (ad, c). Затем он пересылает c A.

– Когда A посылает запрос на выбранный шифротекст (ad, c), то если (ad, c) был записан, Sim_2 отправляет в ответ $\perp$. Иначе Sim_2 посылает испытательный запрос (ad, c). Получив ответ m, он отправляет m A.

– Любые другие запросы и ответы пересылаются без изменений.

Противники удовлетворяют заявленным ограничениям. Первый — поскольку это обеспечено явно. Второй — поскольку симулятор Sim_2 и $\mathrm{Exp}^{\mathrm{int}}_{\Sigma}$ полностью симулируют эксперимент неотличимости. Имеется некоторая пересылка, но для любого нетривиального противника A эта накладная стоимость незначительна.

Начнём с определения некоторых событий. Пусть E_0 — событие $b = b'$, когда B_2 взаимодействует с экспериментом целостности, а E_1 — событие $b = b'$, когда B_1 взаимодействует с экспериментом неотличимости. Заметим, что

$$\mathrm{Adv}^{\text{ind-cpa}}_{\Sigma}(B_1) = 2|\Pr[E_1] - 1/2|.$$

Аналогично, пусть F_0 — событие, что шифротекст, отправленный как испытательный запрос, одновременно расшифровывается в сообщение (не $\perp$) и отсутствует в множестве шифротекстов C эксперимента. Пусть F_1 — событие, что B_2 отвечает $\perp$ на выбранный шифротекст c, но $D(k, c) \neq \perp$. Заметим, что

$$\mathrm{Adv}^{\text{int-ctxt}}_{\Sigma}(B_2) = \Pr[F_0].$$

Наконец, пусть E — событие $b = b'$, когда A взаимодействует с экспериментом неотличимости. Тогда

$$\mathrm{Adv}^{\text{ind-cca}}_{\Sigma}(A) = 2|\Pr[E] - 1/2|.$$

Теперь проводится анализ этих событий. Сначала видно, что F_0 и F_1 — соответствующие события, так что $\Pr[F_0] = \Pr[F_1]$. Далее, если рассмотреть полное взаимодействие B_1 и B_2 с их соответствующими экспериментами, единственное различие состоит в том, что B_2 может отклонить некоторые дополнительные шифротексты, что и есть F_1. При условии, что F_1 никогда не происходит, две игры ведут себя одинаково, что означает

$$\Pr[E_0 \mid \neg F_0] = \Pr[E_1 \mid \neg F_1].$$

При непосредственной проверке видно, что

$$\Pr[E] = \Pr[E_0].$$

Что означает

$$\mathrm{Adv}^{\text{ind-cca}}_{\Sigma}(A) = 2|\Pr[E_0] - 1/2| = 2|\Pr[E_0] - \Pr[E_1] + \Pr[E_1] - 1/2|$$

$$\leq 2|\Pr[E_0] - \Pr[E_1]| + 2|\Pr[E_1] - 1/2|.$$

По предыдущим рассуждениям можно применить Лемму 1.2 и получить

$$|\Pr[E_0] - \Pr[E_1]| \leq \Pr[F_0],$$

и утверждение следует из этого. □

Этот результат наглядно показывает, почему для доказательств необходима целостность шифротекста. Противник при выбранном открытом тексте симулирует ответы на запросы о выбранном шифротексте, отклоняя испытательные шифротексты и просто повторяя запрос, если шифротекст был получен в результате запроса на выбранный открытый текст. Такая симуляция работает при наличии целостности шифротекста. Она не работала бы при целостности открытого текста.

Безвредная изменяемость. Симметричная криптосистема из Листинге 9 тривиально не обладает целостностью шифротекста, даже если базовая криптосистема обладает целостностью шифротекста. Однако легко распознать, что шифротекст (c, i) — тривиальная подделка, поскольку $(c, 1-i)$ должен был быть возвращён в ответ на запрос на выбранный открытый текст. Это часто называется *безвредной изменяемостью*. Хотя это нежелательно в некоторых ситуациях, возникающие технические трудности в доказательствах часто можно обойти, как показывает следующее упражнение.

Неизменяемость и увеличение размера шифротекста. Одной мерой, исторически считавшейся важной и всё ещё важной в некоторых контекстах, является увеличение размера шифротекста, определяемое как разница в длине (обычно в битах) между сообщением и его шифрованием.

Для длинных сообщений увеличение размера шифротекста обычно относительно мало и не играет существенной роли. Однако для систем, обменивающихся большим количеством коротких сообщений, увеличение размера может быть относительно большим (или даже огромным!). Существуют также системы, в которых увеличение размера было бы невозможным, например при скрытом внедрении шифрования в систему, изначально не предназначенную для него.

Одним простым результатом является то, что при наших определениях конфиденциальности детерминированное шифрование небезопасно. Недетерминированное шифрование требует увеличения размера шифротекста. Кроме того, целостность также требует дополнительного увеличения. Это означает, что если увеличение размера недопустимо, необходимо иметь детерминированное шифрование, и целостность при этом невозможна.

Неотличимость достижима, пока противник запрашивает сообщение не более одного раза. Более того, можно иметь свойство, что любое изменение шифротекста вызывает непредсказуемое изменение результата расшифрования. Это называется *неизменяемостью*. В этом случае запросы на выбранный шифротекст не должны сообщать противнику ничего нового.

Для сообщений фиксированной длины детерминированная схема шифрования без увеличения размера шифротекста фактически является биекцией между множеством сообщений и множеством шифротекстов. Если множества сообщений и шифротекстов совпадают, алгоритмы шифрования и расшифрования должны просто вычислять взаимно обратные перестановки.

E

Упражнение 1.9

Рассматриваются симметричные криптосистемы с множеством шифротекстов C, и пусть R — любое отношение на C, вычислимое за полиномиальное время.

(a) Определить R-целостность шифротекста как вариант целостности шифротекста, где испытательный шифротекст принимается как подделка только если он не находится в отношении R ни с одним шифротекстом, возвращённым в ответ на запрос на выбранный открытый текст.

(b) Определить R-неотличимость как вариант неотличимости, где эксперимент также отклоняет запрос на выбранный шифротекст, если шифротекст находится в отношении R с любым испытательным шифротекстом.

(c) Сформулировать и доказать версию Теоремы 1.1 для R-целостности шифротекста и R-неотличимости.

Иными словами, криптосистема определяет семейство перестановок, индексированное длиной сообщения. Криптосистема без увеличения размера шифротекста является семейством индексированных семейств перестановок, зависящим от ключа. Это расширяет понятие блочных шифров. Это направление изучаться далее не будет.

1.6 Шифрование с аутентификацией и ассоциированными данными

Мы видели, что случайность является жизненно важной для безопасного шифрования. Ошибочная генерация случайности представляет серьёзную угрозу в криптографии. Плохая (псевдо-)случайная генерация уже приводила к множеству катастроф, и, по-видимому, имели место также преднамеренные попытки саботировать генераторы псевдослучайных чисел.

Поэтому полезно создавать криптосистемы, которые более устойчивы к неправильному использованию (или более дружелюбны к пользователю, где под пользователем понимается разработчик систем, использующих криптографию), так чтобы при отказе генерации случайности криптография не выходила из

строя полностью. Для этого требуется другой криптографический объект. Мы подчёркиваем, что это объект, который может быть использован для построения симметричной криптосистемы, но сам по себе симметричной криптосистемой не является.

Возникает соблазн построить игру безопасности типа real-or-random для AEAD, но это не работает. Когда противник может указывать одноразовое число, функция шифрования становится детерминированной. Противник мог бы просить шифрования нескольких сообщений очень маленькой длины при фиксированных одноразовом числе и ассоциированных данных и ожидать коллизии, если бы случайные сообщения шифровались. Игра неотличимости может быть сделана работоспособной, однако предпочтительным понятием безопасности для AEAD является свойство, аналогичное случайно-выглядящим шифротекстам из раздела 1.3.

D

Определение 1.7

Схема аутентифицированного шифрования с ассоциированными данными (AEAD) состоит из множества ключей K, множества открытых текстов P, множества ассоциированных данных F, множества одноразовых чисел N и множества шифротекстов C, а также:

- детерминированного алгоритма шифрования E, который по входу ключа, одноразового числа, ассоциированных данных и открытого текста выводит шифротекст;
- детерминированного алгоритма расшифрования D, который по входу ключа, одноразового числа, ассоциированных данных и шифротекста выводит открытый текст либо символ $\perp$.

Для любого ключа k, одноразового числа no, ассоциированных данных ad и открытого текста m выполнено

$$D(k, no, ad, E(k, no, ad, m)) = m.$$

Эксперимент R-rnd-aead $\mathsf{Exp}_{\Sigma}^{\text{R-rnd-aead}}$ выполняется следующим образом:

а) Выбрать ключ $k \xleftarrow{r} K$.

б) Сэмплировать бит $b \xleftarrow{r} \{0,1\}$ и положить $C \leftarrow \varnothing$.

в) Когда противник посылает запрос (no, ad, m) (испытательный), выполнить:

1) Если $(no, ad, m, c) \in C$ для некоторого c, отправить c противнику и прекратить обработку.
2) Если $b = 0$, вычислить $c \leftarrow E(k, no, ad, m)$.
3) Если $b = 1$, сэмплировать $c \xleftarrow{r} R(\ell)$, где ℓ — длина m.
4) Обновить $C \leftarrow C \cup \{(no, ad, m, c)\}$.
5) Отправить c противнику.

г) Когда противник посылает запрос (no, ad, c) (выбранный шифротекст), выполнить:

1) Если $(no, ad, m, c) \in C$ для некоторого m, отправить m противнику.
2) Иначе, если $b = 0$, отправить $D(k, no, ad, c)$ противнику.
3) Иначе, если $b = 1$, отправить $\perp$ противнику.

В конце противник выводит бит $b' \in \{0,1\}$.

Листинг 10 – Эксперимент для игры R-random для AEAD-криптосистемы $\Sigma = (K, P, C, F, N, E, D)$, где R — семейство шума на C.

D

Определение 1.8

Пусть R — семейство шума на C. (τ, le, ld)-противник против R-random безопасности AEAD-криптосистемы Σ с множеством шифротекстов C — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 10, делая не более le запросов выбранного открытого текста и ld запросов выбранного шифротекста, причём время работы противника и эксперимента составляет не более τ.

Преимущество противника определяется как

$$\mathsf{Adv}_{\Sigma}^{R\text{-rnd-aead}}(A) = 2|\Pr[E] - \tfrac{1}{2}|,$$

где E — событие, что бит b', выведенный A, совпадает с b эксперимента.

AEAD легко преобразуется в симметричную криптосистему. Достаточно сэмплировать случайное одноразовое число и зашифровать сообщение, используя это число. Заметим, что шифротекст, выдаваемый алгоритмом шифрования AEAD, должен сопровождаться одноразовым числом, поскольку без него получатель не сможет расшифровать.

Существуют и другие способы выбора одноразового числа или варьирования ассоциированных данных для достижения необходимого эффекта. Например, в диалоге ассоциированные данные могут быть просто счётчиком сообщений, что исключило бы необходимость в одноразовом числе.

Упражнение 1.10

Опишите симметричную криптосистему с ассоциированными данными, которую мы получаем из AEAD-системы при выборе одноразовых чисел случайным образом. Докажите, что для любого противника против конфиденциальности или целостности существует противник против AEAD-схемы с примерно тем же преимуществом, с точностью до малой константы, а для конфиденциальности также с учётом вероятности коллизии случайно выбранных одноразовых чисел.

Упражнение 1.11

Постройте график вероятности коллизии одноразовых чисел для $|N| = 2^{32}, 2^{48}, 2^{64}, 2^{96}, 2^{128}$ и $le = 2, 2^2, 2^3, \ldots, 2^{|N|/2}$. Каково максимальное значение le, если преимущество противника должно быть не более 2^{-20}?

1.7 Несколько ключей

В практике система, использующая симметричные криптосистемы, вряд ли будет ограничивать себя одним ключом. Обычно имеется огромное количество ключей, даже если количество пользователей не велико. Поэтому изучение систем с более чем одним ключом является важным.

Возможно определить варианты игр безопасности, где эксперимент имеет несколько независимых ключей, и противник может выбирать, какой ключ эксперимент должен использовать при обработке запроса.

Как обычно, эти многоключевые понятия содержат одноключевые понятия как частные случаи. В обратную сторону можно доказать, что любого противника против многоключевых понятий можно преобразовать в противника против одноключевого понятия, и преимущество многоключевого противника не превосходит преимущества одноключевого противника, умноженного на количество ключей.

E

Упражнение 1.12

Определить многоключевой вариант ror-cca, сформулировать точный вариант приведённого выше неформального утверждения и с помощью гибридного довода доказать утверждение.

Другой многоключевой вариант заключается в разрешении раскрытия ключей, когда противник может узнать подмножество ключей, выбираемое адаптивно. Непосредственная проблема заключается в том, что противник не может сначала запросить любые испытательные шифротексты под некоторым ключом, а затем позже запросить сам ключ, поскольку это немедленно раскроет испытательный бит. Основная проблема в том, что раскрытие шифротекстов фиксирует эксперимент на определённом ключе, что трудно раскрыть. Большинство естественных обобщений теорем, доказанных для одноключевого случая, трудно доказать для многоключевого случая с компрометацией ключа. Одним из подходов к достижению многоключевой безопасности при компрометации ключа является stateful-шифрование, которое будет исследовано позже.

2 Конфиденциальность и базовые примитивы

Предыдущий раздел касался только смысла безопасности для симметрических криптосистем (и расширял наше понятие симметрических криптосистем). В этом разделе будет рассмотрено, как строить симметрические криптосистемы, обеспечивающие конфиденциальность, и какие примитивы необходимы для их построения. В силу Теоремы 7.8 нужно учитывать только атаки с выбранным открытым текстом (и, следовательно, на данный момент можно также игнорировать ассоциированные данные). Интегритет будет достигнут в следующем разделе.

2.1 Потоковые шифры

Здесь будут рассмотрены так называемые синхронные или аддитивные потоковые шифры, где *генератор потока ключей* расширяет ключ и вектор инициализации в строку символов, которая затем добавляется к сообщению (которое интерпретируется как строка символов). Традиционно потоковые шифры были бит-ориентированными, но алфавит можно понимать как любую группу.

Эксперимент для генератора потока ключей выполняется следующим образом:

1. Сэмплируются $b \xleftarrow{r} \{0,1\}$ и $k \xleftarrow{r} K$.
2. Когда противник посылает запрос длины $l < N$, эксперимент сэмплирует $iv \xleftarrow{r} V$, вычисляет

$$f(k, iv, l) = (z_{0,1}, z_{0,2}, \ldots, z_{0,l}),$$

   и сэмплирует

$$(z_{1,1}, z_{1,2}, \ldots, z_{1,l})$$

   из равномерного распределения на G^l.
3. Затем эксперимент посылает $(iv,\ z_{b,1}, z_{b,2}, \ldots, z_{b,l})$ противнику A.

В конце противник выводит бит $b' \in \{0,1\}$.

Листинг 11 – Эксперимент для игры безопасности генератора потока ключей $f : K \times V \to G^N$.

D

Определение 2.1

Пусть $f : K \times V \to G^N$ является генератором потока ключей. (τ, ℓ_c)-противник против f — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 11, делая не более ℓ_c запросов к эксперименту, и где время работы противника и эксперимента не превосходит τ.

Преимущество этого противника определяется как

$$\mathsf{Adv}_f^{\mathsf{ksg}}(A) = 2|\Pr[E] - 1/2|,$$

где E — это событие, что бит b' на выходе A равен биту b, выбранному экспериментом.

Будет вычисляться только столько элементов потока ключей, сколько требуется. Поток ключей должен вычисляться некоторым алгоритмом, стоимость которого по существу линейна по числу вычисленных элементов потока.

Замечание. Иногда требуется псевдослучайный генератор $f : K \to G^N$. Поскольку вектора инициализации нет, каждый ключ расширяется в единственный поток ключей. Игра безопасности в этом случае является однозапросной версией игры безопасности генератора потока ключей.

Замечание. Существует более сильное понятие безопасности для генераторов потока ключей, где противнику разрешено задавать используемый вектор инициализации (псевдослучайная функция). Обычно это слишком сильное требование, поскольку оно не нужно и может усложнить проектирование генератора потока ключей. Промежуточным вариантом является указание фиксированной последовательности векторов инициализации, для которой легко проектировать и которая полезна во многих приложениях.

Пример 2.1

Пусть $f : K \times V \to G^N$ является генератором потока ключей. Аддитивный потоковый шифр, основанный на f, имеет вид

$$\Sigma = (K,\ \bigcup_{l \le N} G^l,\ V \times \bigcup_{l \le N} G^l,\ E,\ D),$$

где:

– Алгоритм шифрования E принимает на вход ключ $k \in K$ и сообщение $m = m_1 m_2 \dots m_l$ для некоторого $l \le N$. Он сэмплирует $iv \xleftarrow{r} V$, вычисляет

$$(z_1, z_2, \dots, z_l) \leftarrow f(k, iv, l),$$

и $w_i = m_i + z_i$ для $i = 1, 2, \dots, l$, после чего выводит $c = (iv,\ w_1 w_2 \dots w_l)$.

– Алгоритм расшифрования D принимает на вход ключ $k \in K$ и шифротекст $c = (iv,\ w_1 w_2 \dots w_l)$. Он вычисляет

$$(z_1, z_2, \dots, z_l) \leftarrow f(k,\ iv,\ l),$$

и $m_i = w_i - z_i$ для всех $i = 1, 2, \dots, l$, после чего выводит $m = m_1 m_2 \dots m_l$.

Упражнение 2.1

Пусть Σ — аддитивный потоковый шифр, основанный на генераторе потока ключей $f : K \times V \to G^N$. Пусть A — $(\tau,\ \ell_c,\ \ell_e,\ 0)$-противник против неразличимости для Σ. Показать, что существует $(\tau',\ \ell_c + \ell_e)$-противник B против генератора потока ключей f, где τ' по существу равен τ, и

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\Sigma}(A) \le \mathsf{Adv}^{\mathsf{ksg}}_{f}(B) + 2\,\frac{(\ell_c + \ell_e)^2}{|V|}.$$

Замечание. Дополнительный член в границе преимущества противника возникает из-за возможных коллизий вектора инициализации. Если использовать более сильные notions безопасности для генератора потока ключей, такие как описанные в замечании перед примером, можно построить потоковые шифры с более жёсткими гарантиями безопасности, используя фиксированную последовательность векторов инициализации. Однако это приводит к stateful-шифру. Иногда это вполне практично, но иногда — нет.

E

Упражнение 2.2

Повторить предыдущее упражнение, но для R-random-безопасности, определённой в Разделе 1.3.

2.2 Псевдослучайные перестановки и функции

Блочные шифры, также называемые псевдослучайными перестановками, являются чрезвычайно полезными примитивами в симметричной криптографии. Блочный шифр представляет собой семейство перестановок на некотором множестве.

Связанным понятием является псевдослучайная функция, которую мы уже видели в ослабленной форме — генератор потока ключей. Псевдослучайная функция (семейство) — это просто семейство функций, индексированное множеством ключей.

D

Определение 2.2

Псевдослучайное функциональное (PRF) семейство — это функция

$$f : K \times S \to T.$$

Идея этих игр безопасности состоит в том, что эксперимент предоставляет противнику доступ либо к функции/перестановке из псевдослучайного семейства, либо к случайной функции/перестановке. Для простоты эксперимент с перестановками вычисляет как перестановку, так и её обратную для каждого запроса.

Эксперимент PRF выполняется следующим образом:

1. Сэмплировать $b \xleftarrow{r} \{0,1\}$ и $k \xleftarrow{r} K$. Положить $C_0 = C_1 = \varnothing$.
2. Когда противник посылает запрос $s \in S$ (evaluate), выполнить:
   a) Если $s \notin C_1$, сэмплировать $u \xleftarrow{r} T$ и добавить s в C_1 и (s, u) в C_0.
   b) Найти $(s, u_1) \in C_0$. Вычислить $u_0 \leftarrow f(k, s)$. Отправить u_b противнику.

В конце противник выводит $b' \in \{0,1\}$.

Листинг 12 – Эксперимент для игры безопасности PRF для псевдослучайной функции $f : K \times S \to T$.

D

Определение 2.3

Пусть $f : K \times S \to T$ — PRF. (τ, l_c)-противник против f — это интерактивный алгоритм A, который взаимодействует с экспериментом на Листинге 12, делая не более l_c запросов к эксперименту, причём время работы противника и эксперимента не превышает τ.
Преимущество этого противника определяется как

$$\mathsf{Adv}_f^{\mathrm{prf}}(A) = 2\left|\Pr[E] - \frac{1}{2}\right|,$$

где E — событие, что бит b' на выходе A совпадает с битом b, выбранным экспериментом.

D

Определение 2.4

Пусть $\pi, \pi^{-1} : K \times S \to S$ образуют блочный шифр. (τ, l_c)-противник против (π, π^{-1}) — это интерактивный алгоритм A, взаимодействующий с экспериментом на Листинге 13, делая не более l_c запросов к эксперименту, причём время работы противника и эксперимента не превышает τ.
Преимущество этого противника определяется как

$$\mathsf{Adv}_{(\pi,\pi^{-1})}^{\mathrm{prp}}(A) = 2\left|\Pr[E] - \frac{1}{2}\right|,$$

где E — событие, что бит b' на выходе A совпадает с битом b, выбранным экспериментом.

Эксперимент для блочного шифра выполняется следующим образом:

1. Сэмплировать $b \xleftarrow{r} \{0,1\}$ и $k \xleftarrow{r} K$. Положить $C_0 = C_1 = C_2 = \varnothing$.
2. Когда противник посылает запрос $s \in S$ (evaluate), выполнить:
   a) Если $s \notin C_1$, сэмплировать $u \xleftarrow{r} S \setminus C_2$ и добавить (s, u) в C_0, s в C_1 и u в C_2. Если $s \notin C_2$, сэмплировать $v \xleftarrow{r} S \setminus C_1$ и добавить (v, s) в C_0, s в C_2 и v в C_1.
   b) Найти $(s, u_1) \in C_0$ и $(v_1, s) \in C$. Вычислить $u_0 \leftarrow \pi^{-1}(k, s)$ и $v_0 \leftarrow \pi(k, s)$. Отправить (u_b, v_b) противнику.

В конце противник выводит $b' \in \{0,1\}$.

Листинг 13 – Эксперимент для игры безопасности блочного шифра (π, π^{-1}) : $K \times S \to S$.

Замечание. Существует более слабое понятие псевдослучайной функции (иногда называемое слабой псевдослучайной функцией), когда противник не выбирает точку, в которой вычисляется функция. Это по сути то же, что генератор потока ключей, за исключением того, что генератор потока ключей имеет маленькую область определения и очень большую область значений, тогда как для псевдослучайных функций часто наоборот. Область значений генератора потока ключей также имеет очень специальную форму, тогда как псевдослучайная функция может иметь более сложное множество значений.

Замечание. Проектирование псевдослучайных функций и перестановок выходит за рамки этой книги. Тем не менее псевдослучайные перестановки или блочные шифры имеют долгую историю. Считается, что известны хорошие блочные шифры.

Псевдослучайные функции получили меньше прямого внимания, хотя современные стандарты хеширования дают некоторые очень полезные разработки. Функции сжатия многих распространённых хеш-функций подходят как псевдослучайные функции для некоторых целей. Конструкция HMAC также полезна, если требуется функция с более крупной областью определения. Другая общая стратегия основана на методе *forthwidth*.

Использование блочных шифров в качестве псевдослучайных функций Псевдослучайная перестановка выглядит как случайная перестановка. Псевдослучайная функция выглядит как случайная функция. Но до тех пор, пока исследуется не слишком много значений, случайная перестановка выглядит как случайная функция. Таким образом, если имеется псевдослучайная перестановка, её можно рассматривать как псевдослучайную функцию, пока она вычисляется не на слишком большом числе точек.

Сэмплирование случайной функции и последующее вычисление её значений на различных точках эквивалентно независимому выбору элементов из равномерного распределения. Точно так же сэмплирование случайной перестановки и вычисление её значений на различных точках эквивалентно независимому выбору элементов из равномерного распределения, с тем лишь условием, что выбранные элементы различны.

Эксперимент по различению случайных функций и случайных перестановок с противником A выполняется следующим образом:

1. Сэмплировать бит $b \xleftarrow{r} \{0,1\}$. Положить $C_0 = C_1 = C_2 = \varnothing$.
2. Когда противник посылает запрос $s \in S$ (evaluate), выполнить:
   a) Если $s \notin C_1$, сэмплировать $u_0 \xleftarrow{r} S$ и $u_1 \xleftarrow{r} S \setminus C_2$. Добавить (s, u_b) в C_0, s в C_1 и u_b в C_2.
   b) Найти $(s, u) \in C_0$. Отправить u противнику A.

В конце противник выводит бит $b' \in \{0,1\}$.

Листинг 14 – Эксперимент для леммы 2.1

Лемма 2.1

Пусть A — интерактивный алгоритм, который взаимодействует с экспериментом из Листинга 14 и делает не более l_c запросов. Тогда

$$2|\Pr[E] - \tfrac{1}{2}| \leq \frac{l_c^2}{2|S|},$$

где E — событие, что $b' = b$ в конце работы противника.

Доказательство. Эксперимент по существу выбирает две последовательности элементов из S, одну с возвращением, а другую без возвращения. Обозначим через F событие, что последовательность, выбранная с возвращением, состоит из различных элементов.

Утверждение о парадоксе дней рождения говорит, что последовательность, выбранная с возвращением, будет состоять из различных элементов, кроме как с вероятностью, ограниченной величиной $l_c^2/(2|S|)$, то есть

$$\Pr[\neg F] \leq \frac{l_c^2}{2|S|}.$$

Если все элементы, выбранные с возвращением, различны, то они распределены точно так же, как элементы, выбранные без возвращения, следовательно,

$$\Pr[E \mid F] = \tfrac{1}{2}.$$

Вычисляется:

$$2|\Pr[E] - \tfrac{1}{2}| = 2\big|\Pr[E \mid F](1 - \Pr[\neg F]) + \Pr[E \mid \neg F]\Pr[\neg F] - \tfrac{1}{2}\big|$$

$$= 2|(\Pr[E \mid \neg F] - \tfrac{1}{2})\Pr[\neg F]| \leq \Pr[\neg F].$$

Тем самым утверждение доказано. □

Лемма выше показывает, что пока вычисления выполняются не на слишком многих точках, случайно выбранная функция неотличима от случайно выбранной перестановки. Поскольку блочный шифр должен быть трудноотличим от случайной перестановки (с инверсией), а псевдослучайная функция — от случайной функции, то по транзитивности следует, что блочный шифр (если не использовать его обратную функцию) должен быть трудноотличим от случайной функции.

Теорема 2.1

(Лемма о переключении между PRP и PRF) Пусть (π, π^{-1}) — блочный шифр на множестве S. Пусть A — (τ, l_c)-противник против π, рассматриваемого как PRF. Тогда существует (τ', l_c)-противник B против (π, π^{-1}), рассматриваемого как блочный шифр, где τ' по существу равен τ, и

$$\mathsf{Adv}^{\mathrm{prf}}_{\pi}(A) \;\leq\; \mathsf{Adv}^{\mathrm{prp}}_{(\pi,\pi^{-1})}(B) \;+\; \frac{l_c^2}{2|S|}.$$

Доказательство. Сначала определяется симулятор Sim следующим образом: когда симулятор получает запрос от A, он пересылает запрос в эксперимент блочного шифра. Когда он получает ответ (u', u''), он пересылает u' алгоритму A. Противник B запускает копию Sim и A, и когда A выводит бит b', B также выводит b'. Если A превышает свои ограничения, B сэмплирует $b' \xleftarrow{r} \{0,1\}$ и выводит его.

Далее вводятся события. Пусть E_β — событие, что A выводит 1 при взаимодействии с PRF-экспериментом, где $b = \beta$. Пусть F_β — событие, что B выводит 1 при взаимодействии с экспериментом блочного шифра, где $b = \beta$.

По Лемме 1.1:

$$\mathsf{Adv}^{\mathrm{prf}}_{\pi}(A) = |\Pr[E_0] - \Pr[E_1]|, \qquad \mathsf{Adv}^{\mathrm{prp}}_{(\pi,\pi^{-1})}(B) = |\Pr[F_0] - \Pr[F_1]|.$$

По непосредственной проверке, когда $b = 0$, поведение A во взаимодействии с PRF-экспериментом полностью совпадает с поведением B во взаимодействии с экспериментом блочного шифра. Следовательно,

$$\Pr[E_0] = \Pr[F_0].$$

Далее, также по непосредственной проверке, поведение A во взаимодействии с PRF-экспериментом при $b = 1$ совпадает с поведением A во взаимодействии с экспериментом из Листинга 14 при $b = 0$. Аналогично, поведение A, встроенного в B, при взаимодействии с экспериментом блочного шифра при $b = 1$ совпадает с поведением A во взаимодействии с экспериментом из Листинга 14 при $b = 1$. Леммы 1.1 и 2.1 дают:

$$|\Pr[E_1] - \Pr[F_1]| \ \leq \ \frac{l_c^2}{2|S|}.$$

Наконец, вычисляется:

$$\mathsf{Adv}_{\pi}^{\mathrm{prf}}(A) = |\Pr[E_0] - \Pr[E_1]|$$

$$= |\Pr[E_0] - \Pr[F_0] + \Pr[F_0] - \Pr[F_1] + \Pr[F_1] - \Pr[E_1]|$$

$$\leq |\Pr[F_0] - \Pr[F_1]| \ + \ |\Pr[F_1] - \Pr[E_1]|.$$

Таким образом, утверждение доказано. □

2.3 Два построения

Ранее был рассмотрен режим счётчика (counter mode) для блочных шифров, но эта конструкция одинаково хорошо работает и для PRF.

Пример 2.2

Пусть f — функция из $K \times V \times \{1, 2, \ldots, N\}$ в некоторую конечную группу G. Для любых $iv \in V$ и $k \in K$ режим счётчика (CTR) определяется функцией

$$\mathrm{ctr}_\pi : K \times V \to G^N, \qquad \mathrm{ctr}_\pi(k, iv) = z_1 z_2 \ldots z_N,$$

где

$$z_i = \pi(k, (iv, i)), \qquad 1 \leq i \leq N.$$

Упражнение 2.3

Доказать, что любого противника против генератора ключевого потока в режиме счётчика можно превратить в PRF-противника против f с той же выгодой и практически тем же временем выполнения.

Замечание. Приведённая конструкция по существу показывает, как построить PRF с более крупным ко-доменом ценой уменьшения домена. Позднее будет рассмотрено, как увеличить домен PRF.

Режим сцепления блоков (cipher-block chaining, CBC) из Примера 1.9 — удобный способ превратить блочный шифр в симметрическую криптосистему.

Упражнение 2.4

Пусть (π, π^{-1}) — блочный шифр над группой G. Пусть R — семейство шумовых распределений, где R_ℓ — равномерное распределение на $G^{\ell+1}$. Доказать, что любого R-random противника с выбранным открытым текстом против режима CBC можно превратить в противника против блочного шифра с практически тем же временем выполнения. Если l_c — число зашифрованных сообщений, а l — их суммарная длина, то разность выгод ограничена величиной

$$\frac{(l_c + l)^2}{|G|}.$$

ЗАКЛЮЧЕНИЕ

Проведенное исследование позволило систематизировать и углубить понимание безопасности симметричных криптосистем. Основные достижения работы включают:

а) **Формализацию понятий безопасности**. Были строго определены и сопоставлены различные подходы к конфиденциальности (семантическая безопасность, неотличимость, real-or-random), доказана их эквивалентность. Это предоставляет гибкий инструментарий для анализа криптосистем.

б) **Анализ целостности**. Показано, что целостность шифротекста является более сильным и полезным свойством, чем целостность открытого текста, особенно в контексте доказательств безопасности при атаках с выбранным шифротекстом.

в) **Исследование криптографических примитивов**. Изучены свойства потоковых шифров, PRF и PRP, установлены связи между ними. Лемма о переключении между PRP и PRF позволяет использовать блочные шифры в конструкциях, требующих псевдослучайных функций.

г) **Практические конструкции**. Режимы CTR и CBC проанализированы с точки зрения безопасности, получены оценки преимущества противника в зависимости от параметров системы.

д) **Методологический вклад**. Работа демонстрирует мощь формальных методов в криптографии, включая редукционные доказательства, гибридные аргументы и симуляционные техники.

Проведенная работа закладывает фундамент для дальнейшего изучения и разработки безопасных симметричных криптосистем, сочетая теоретическую строгость с практической применимостью.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Gjøsteen K. Practical Mathematical Cryptography. — 1st Edition. — New York : Chapman, Hall/CRC, 2023. — С. 546. — ISBN 9781003149422. — DOI: 10.1201/9781003149422.