

Содержание

ПРЕДИСЛОВИЕ	29
ПРЕДИСЛОВИЕ Уитфилда Диффи	33
ВВЕДЕНИЕ	39
КАК ЧИТАТЬ ЭТУ КНИГУ	40
БЛАГОДАРНОСТИ	43
ОБ АВТОРЕ	45
ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ	47
1.1. ТЕРМИНОЛОГИЯ	47
Отправитель и получатель	47
Сообщения и шифрование	47
Аутентификация, целостность и неотрицание авторства	48
Алгоритмы и ключи	49
Симметричные алгоритмы	50
Алгоритмы с открытым ключом	51
Криптоанализ	52
Безопасность алгоритмов	56
Исторические термины	57
1.2. СТЕГАНОГРАФИЯ	58
1.3. ПОДСТАНОВОЧНЫЕ И ПЕРЕСТАНОВОЧНЫЕ ШИФРЫ	58
Подстановочные шифры	59
Перестановочные шифры	61
Роторные машины	62
Для дальнейшего чтения	63
1.4. ПРОСТАЯ ОПЕРАЦИЯ ХОР	63
1.5. ОДНОРАЗОВЫЕ БЛОКНОТЫ	65
1.6. КОМПЬЮТЕРНЫЕ АЛГОРИТМЫ	68
1.7. БОЛЬШИЕ ЧИСЛА	69

Часть I

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ 71

ГЛАВА 2. СТРУКТУРНЫЕ ЭЛЕМЕНТЫ ПРОТОКОЛОВ	73
2.1. ВВЕДЕНИЕ В ПРОТОКОЛЫ	73
Предназначение протоколов	75
Действующие лица	75
Протоколы с посредником	76
Арбитражные протоколы	79
Самодостаточные протоколы	80
Атаки на протоколы	80

2.2. ОБМЕН СООБЩЕНИЯМИ С ПОМОЩЬЮ СИММЕТРИЧНОЙ КРИПТОГРАФИИ	81
2.3. ОДНОСТОРОННИЕ ФУНКЦИИ	83
2.4. ОДНОСТОРОННИЕ ХЕШ-ФУНКЦИИ	84
Коды проверки подлинности сообщения	86
2.5. ОБМЕН СООБЩЕНИЯМИ С ПОМОЩЬЮ КРИПТОГРАФИИ С ОТКРЫТЫМ КЛЮЧОМ	86
Гибридные криптосистемы	88
Головоломки Меркла	90
2.6. ЦИФРОВЫЕ ПОДПИСИ	91
Подпись документа с помощью симметричных криптосистем и посредника	92
Деревья цифровых подписей	94
Подпись документа с помощью криптографии с открытым ключом	94
Подпись документа и метки времени	95
Подписание документов с помощью криптографии с открытым ключом и односторонних хеш-функций	96
Алгоритмы и терминология	97
Множественные подписи	97
Невозможность отказа от авторства и цифровые подписи	98
Применение цифровых подписей	99
2.7. ЦИФРОВЫЕ ПОДПИСИ И ШИФРОВАНИЕ	99
Возвращение полученного сообщения	101
Отражение атаки, основанной на повторной пересылке сообщений	102
Атаки криптосистем с открытыми ключами	103
2.8. ГЕНЕРАЦИЯ СЛУЧАЙНЫХ И ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ	103
Псевдослучайные последовательности	104
Криптографически стойкие псевдослучайные последовательности	105
Истинно случайные последовательности	106
ГЛАВА 3. ОСНОВНЫЕ ПРОТОКОЛЫ	107
3.1. ОБМЕН КЛЮЧАМИ	107
Обмен ключами с помощью симметричной криптографии	107
Обмен ключами с помощью криптографии с открытым ключом	108
Атака “человек посередине”	108
Протокол взаимоблокировки	109
Обмен ключами с помощью цифровых подписей	110
Одновременная передача ключей и сообщений	111
Широковещательная рассылка ключей и сообщений	112

3.2. АУТЕНТИФИКАЦИЯ	113
Аутентификация с помощью односторонних функций	113
Атака по словарю и “соль”	113
Программа SKEY	114
Аутентификация с помощью криптографии с открытым ключом	115
Взаимная аутентификация с помощью протокола взаимоблокировки	116
Протоколы SKID	117
Аутентификация сообщений	118
3.3. АУТЕНТИФИКАЦИЯ И ОБМЕН КЛЮЧАМИ	119
Протокол Wide-Mouth Frog	119
Протокол Yahalom	120
Протокол Нидхема—Шредера	121
Протокол Отвея—Рииса	123
Протокол Kerberos	123
Протокол Ньюмана—Стаблбайна	124
Протокол DASS	126
Протокол Деннинга—Сакко	127
Протокол Ву—Лама	128
Другие протоколы	129
Выводы	129
3.4. ФОРМАЛЬНЫЙ АНАЛИЗ ПРОТОКОЛОВ АУТЕНТИФИКАЦИИ И ОБМЕНА КЛЮЧАМИ	130
3.5. КРИПТОГРАФИЯ С НЕСКОЛЬКИМИ ОТКРЫТЫМИ КЛЮЧАМИ	134
Широковещательная передача сообщения	135
3.6. РАЗБИЕНИЕ СЕКРЕТА	136
3.7. РАЗДЕЛЕНИЕ СЕКРЕТА	138
Разделение секрета с мошенниками	139
Разделение секрета без помощи Трента	140
Разделение секрета без раскрытия долей	140
Верифицированное разделение секрета	140
Схемы разделения секрета с предохранительными мерами	141
Разделение секрета с вычеркиванием из списка	141
3.8. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА БАЗ ДАННЫХ	141
ГЛАВА 4. ПРОМЕЖУТОЧНЫЕ ПРОТОКОЛЫ	143
4.1. СЛУЖБЫ МЕТОК ВРЕМЕНИ	143
Решение с посредником	143
Улучшенный протокол с посредником	144
Протокол связывания	145
Распределенный протокол	146

Дальнейшая работа	147
4.2. СКРЫТЫЙ КАНАЛ	147
Применения скрытого канала	149
Подписи, свободные от скрытого канала	150
4.3. НЕОСПОРИМЫЕ ЦИФРОВЫЕ ПОДПИСИ	150
4.4. ПОДПИСИ, ПОДТВЕРЖДАЕМЫЕ ДОВЕРЕННЫМИ ЛИЦАМИ	152
4.5. ПОДПИСИ ПО ДОВЕРЕННОСТИ	153
4.6. ГРУППОВЫЕ ПОДПИСИ	154
Групповые подписи с доверенным посредником	155
4.7. ПОДПИСИ С ОБНАРУЖЕНИЕМ ПОДДЕЛКИ	155
4.8. ВЫЧИСЛЕНИЯ НАД ЗАШИФРОВАННЫМИ ДАННЫМИ	157
4.9. ПЕРЕДАЧА БИТОВ	157
Передача битов с помощью симметричной криптографии	158
Передача бита с помощью односторонних функций	159
Передача бита с помощью генератора псевдослучайной последовательности	159
Двоичные объекты	160
4.10. ЖЕРЕБЬЕВКА С ПОМОЩЬЮ ИДЕАЛЬНОЙ МОНЕТЫ	161
Жеребьевка с помощью односторонних функций	162
Жеребьевка с помощью криптографии с открытым ключом	162
Бросок монеты в колодец	164
Генерация ключей с помощью жеребьевки	164
4.11. МЫСЛЕННЫЙ ПОКЕР	164
Мысленный покер с тремя игроками	165
Атаки на протоколы мысленного покера	167
Анонимное распределение ключей	167
4.12. ОДНОСТОРОННИЕ СУММАТОРЫ	169
4.13. РАСКРЫТИЕ СЕКРЕТОВ ПО ПРИНЦИПУ “ВСЕ ИЛИ НИЧЕГО”	170
4.14. ДЕПОНИРОВАНИЕ КЛЮЧЕЙ	171
Стратегии депонирования	173
ГЛАВА 5. УСОВЕРШЕНСТВОВАННЫЕ ПРОТОКОЛЫ	177
5.1. ДОКАЗАТЕЛЬСТВА С НУЛЕВЫМ ЗНАНИЕМ	177
Базовый протокол с нулевым разглашением	178
Изоморфизм графа	181
Гамильтоновы циклы	182
Параллельные доказательства с нулевым разглашением	183
Неинтерактивные доказательства с нулевым разглашением	184

Общие замечания	186
5.2. ИСПОЛЬЗОВАНИЕ ДОКАЗАТЕЛЬСТВА С НУЛЕВЫМ РАЗГЛАШЕНИЕМ	
для идентификации	187
Проблема гроссмейстера	188
Мошенничество мафии	188
Обман, осуществленный террористами	189
Предлагаемые решения	189
Обман с несколькими лицами	190
Прокат паспортов	190
Доказательство членства	191
5.3. СЛЕПЫЕ ПОДПИСИ	191
Полностью слепые подписи	191
Слепые подписи	192
Патенты	195
5.4. ЛИЧНОСТНАЯ КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ	195
5.5. ЗАБЫВЧИВАЯ ПЕРЕДАЧА	196
5.6. ЗАБЫВЧИВЫЕ ПОДПИСИ	198
5.7. ОДНОВРЕМЕННОЕ ПОДПИСАНИЕ КОНТРАКТА	199
Подпись контракта с помощью посредника	199
Одновременная подпись контракта без посредника (при личной встрече)	200
Одновременная подпись контракта без посредника (без личной встречи)	200
Одновременная подпись контракта без посредника (с помощью криптографии)	202
5.8. ЗАКАЗНАЯ ЭЛЕКТРОННАЯ ПОЧТА	204
5.9. ОДНОВРЕМЕННЫЙ ОБМЕН СЕКРЕТАМИ	207
Глава 6. ЭЗОТЕРИЧЕСКИЕ ПРОТОКОЛЫ	209
6.1. ТАЙНОЕ ГОЛОСОВАНИЕ	209
Упрощенный протокол голосования №1	209
Упрощенный протокол голосования №2	210
Голосование со слепыми подписями	210
Голосование с двумя центральными комиссиями	212
Голосование с одной центральной комиссией	213
Улучшенное голосование с одной центральной комиссией	214
Голосование без центральной избирательной комиссии	216
Другие схемы голосования	220
6.2. СЕКРЕТНЫЕ МНОГОСТОРОННИЕ ВЫЧИСЛЕНИЯ	221
Протокол №1	221
Протокол №2	222

Протокол №3	223
Протокол №4	224
Безусловно тайные многосторонние протоколы	225
Тайное вычисление схемы	225
6.3. ШИРОКОВЕЩАТЕЛЬНАЯ ПЕРЕДАЧА АНОНИМНЫХ СООБЩЕНИЙ	225
6.4. ЭЛЕКТРОННЫЕ ДЕНЬГИ	228
Протокол №1	229
Протокол №2	230
Протокол №3	231
Протокол №4	232
Электронные деньги и идеальное преступление	236
Реальные электронные наличные	236
Другие протоколы электронных денег	236
Анонимные кредитные карточки	238

Часть II

МЕТОДЫ КРИПТОГРАФИИ 241

Глава 7. Длина ключа	243
7.1. Длина симметричного ключа	243
Оценка продолжительности и стоимости лобовой атаки	244
Программы для взлома	247
Нейронные сети	248
Вирусы	248
Китайская лотерея	249
Биотехнология	250
Термодинамические ограничения	251
7.2. Длина открытого ключа	252
Вычисление с помощью ДНК	259
Квантовые вычисления	261
7.3. Сравнение длин симметричных и открытых ключей	262
7.4. Атака на основе парадокса дней рождения и односторонние хеш-функции	263
7.5. Какой должна быть длина ключа?	263
7.6. Предостережение	265
Глава 8. Управление ключами	267
8.1. Генерация ключей	268
Уменьшенные пространства ключей	268
Неправильный выбор ключей	270
Случайные ключи	272

Ключевые фразы	273
Стандарт генерации ключей X9.17	274
Генерация ключей в Министерстве обороны США	275
8.2. НЕЛИНЕЙНЫЕ ПРОСТРАНСТВА КЛЮЧЕЙ	275
8.3. ПЕРЕСЫЛКА КЛЮЧЕЙ	276
Распределение ключей в крупных сетях	278
8.4. ПРОВЕРКА КЛЮЧЕЙ	278
Обнаружение ошибок при пересылке ключей	280
Обнаружение ошибок при расшифровке	280
8.5. ИСПОЛЬЗОВАНИЕ КЛЮЧЕЙ	281
Контроль использования ключей	282
8.6. ОБНОВЛЕНИЕ КЛЮЧЕЙ	282
8.7. ХРАНЕНИЕ КЛЮЧЕЙ	283
8.8. РЕЗЕРВНЫЕ КЛЮЧИ	284
8.9. СКОМПРОМЕТИРОВАННЫЕ КЛЮЧИ	285
8.10. СРОК ДЕЙСТВИЯ КЛЮЧЕЙ	286
8.11. РАЗРУШЕНИЕ КЛЮЧЕЙ	288
8.12. УПРАВЛЕНИЕ КЛЮЧАМИ В СИСТЕМАХ С ОТКРЫТЫМ КЛЮЧОМ	289
Сертификаты открытых ключей	290
Распределенное управление ключами	291
Глава 9. ТИПЫ АЛГОРИТМОВ И КРИПТОГРАФИЧЕСКИХ РЕЖИМОВ	293
9.1. РЕЖИМ ЭЛЕКТРОННОЙ КОДОВОЙ КНИГИ	294
Заполнение блоков	295
9.2. ПОВТОР БЛОКА	296
9.3. РЕЖИМ СЦЕПЛЕНИЯ БЛОКОВ ШИФРОТЕКСТА	298
Вектор инициализации	299
Дополнение	300
Распространение ошибки	302
Вопросы безопасности	302
9.4. ПОТОКОВЫЕ ШИФРЫ	303
9.5. САМОСИНХРОНИЗИРУЮЩИЕСЯ ПОТОКОВЫЕ ШИФРЫ	305
Вопросы безопасности	306
9.6. РЕЖИМ ГАММИРОВАНИЯ С ОБРАТНОЙ СВЯЗЬЮ	306
Вектор инициализации	308
Распространение ошибки	309
9.7. СИНХРОННЫЕ ПОТОКОВЫЕ ШИФРЫ	309
Атака вставкой	311
9.8. РЕЖИМ ОБРАТНОЙ СВЯЗИ ПО ВЫХОДУ	311
Вектор инициализации	312

Распространение ошибки	312
Режим OFB и проблемы безопасности	313
Потоковые шифры в режиме OFB	313
9.9. РЕЖИМ СЧЕТЧИКА	314
Потоковые шифры в режиме счетчика	314
9.10. ДРУГИЕ РЕЖИМЫ БЛОЧНЫХ ШИФРОВ	315
Режим сцепления блоков	315
Режим сцепления блоков шифра с распространением ошибки	316
Сцепление блоков шифротекста с контрольной суммой	317
Нелинейная обратная связь по выходу	317
Прочие режимы	317
9.11. ВЫБОР РЕЖИМА ШИФРОВАНИЯ	318
9.12. ЧЕРЕДОВАНИЕ	321
9.13. СРАВНЕНИЕ БЛОЧНЫХ И ПОТОКОВЫХ ШИФРОВ	322
ГЛАВА 10. ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ	325
10.1. ВЫБОР АЛГОРИТМА	326
Экспорт алгоритмов	328
10.2. СРАВНЕНИЕ КРИПТОГРАФИИ С ОТКРЫТЫМ КЛЮЧОМ И СИММЕТРИЧНОЙ КРИПТОГРАФИИ	328
10.3. ШИФРОВАНИЕ КАНАЛОВ СВЯЗИ	330
Канальное шифрование	330
Сквозное шифрование	332
Объединение двух подходов	333
10.4. ШИФРОВАНИЕ ДАННЫХ ДЛЯ ХРАНЕНИЯ	335
Разыменованние ключей	336
Шифрование на файловом уровне и на уровне драйверов	336
Обеспечение произвольного доступа к зашифрованному диску	338
10.5. СРАВНЕНИЕ АППАРАТНОГО И ПРОГРАММНОГО СПОСОБОВ ШИФРОВАНИЯ	339
Аппаратное шифрование	339
Программное шифрование	341
10.6. СЖАТИЕ, КОДИРОВАНИЕ И ШИФРОВАНИЕ	342
10.7. ОБНАРУЖЕНИЕ ЗАШИФРОВАННЫХ ДАННЫХ	342
10.8. СОКРЫТИЕ ШИФРОТЕКСТА В ШИФРОТЕКСТЕ	343
10.9. РАЗРУШЕНИЕ ИНФОРМАЦИИ	345

Часть III
КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ 347

Глава 11. МАТЕМАТИЧЕСКИЕ ОСНОВЫ	349
11.1. ТЕОРИЯ ИНФОРМАЦИИ	349
Энтропия и неопределенность	349
Энтропия языка	350
Стойкость криптосистем	351
Расстояние единственности	352
Практическое использование теории информации	354
Перемешивание и рассеивание	354
11.2. ТЕОРИЯ СЛОЖНОСТИ	355
Сложность алгоритмов	355
Сложность задач	357
NP-полные задачи	360
11.3. ТЕОРИЯ ЧИСЕЛ	361
Модулярная арифметика	361
Простые числа	364
Наибольший общий делитель	365
Обратные значения по модулю	366
Вычисление коэффициентов	368
Малая теорема Ферма	368
Функция Эйлера	369
Китайская теорема об остатках	370
Квадратичные вычеты	371
Символ Лежандра	372
Символ Якоби	373
Целые числа Блюма	374
Образующие	375
Вычисление в поле Гауа	376
11.4. ФАКТОРИЗАЦИЯ ЦЕЛЫХ ЧИСЕЛ	378
Квадратные корни по модулю n	381
11.5. ГЕНЕРАЦИЯ ПРОСТЫХ ЧИСЕЛ	381
Тест Соловея—Штрассена	382
Тест Леманна	383
Тест Рабина—Миллера	383
Практические соображения	384
Сильные простые числа	385
11.6. ДИСКРЕТНЫЕ ЛОГАРИФМЫ В КОНЕЧНОМ ПОЛЕ	386
Вычисление дискретных логарифмов в конечной группе	386

ГЛАВА 12. СТАНДАРТ ШИФРОВАНИЯ ДАННЫХ DES	389
12.1. Основы	389
Разработка стандарта	389
Принятие стандарта	392
Аттестация и сертификация оборудования DES	393
События 1987 года	394
События 1993 года	395
12.2. ОПИСАНИЕ СТАНДАРТА DES	396
Схема алгоритма	396
Начальная перестановка	398
Преобразования ключа	399
Расширяющая перестановка	400
Подстановка с помощью S-блоков	401
Перестановка с помощью P-блоков	404
Заключительная перестановка	404
Расшифровка в алгоритме DES	405
Режимы алгоритма DES	405
Аппаратные и программные реализации DES	405
12.3. СТОЙКОСТЬ АЛГОРИТМА DES	407
Слабые ключи	408
Комплементарные ключи	411
Алгебраическая структура	411
Длина ключа	412
Количество раундов	413
Проектирование S-блоков	414
Дополнительные результаты	415
12.4. ДИФФЕРЕНЦИАЛЬНЫЙ И ЛИНЕЙНЫЙ КРИПТОАНАЛИЗ	415
Дифференциальный криптоанализ	415
Криптоанализ на основе связанных ключей	421
Линейный криптоанализ	422
Дальнейшие направления	425
12.5. ПРАКТИЧЕСКИЕ КРИТЕРИИ ПРОЕКТИРОВАНИЯ	426
12.6. ВАРИАНТЫ АЛГОРИТМА DES	427
Множественный алгоритм DES	427
Алгоритм DES с независимыми подключками	427
Алгоритм DESX	428
Алгоритм CRYPT(3)	428
Обобщенный алгоритм DES	428
Алгоритм DES с измененными S-блоками	430
Алгоритм RDES	430

Алгоритм s^n DES	431
Алгоритм DES с S-блоками, зависящими от ключа	432
12.7. НАСКОЛЬКО СТОЕК АЛГОРИТМ DES В НАСТОЯЩЕЕ ВРЕМЯ?	434
ГЛАВА 13. ДРУГИЕ БЛОЧНЫЕ ШИФРЫ	437
13.1. АЛГОРИТМ LUCIFER	437
13.2. АЛГОРИТМ MADRYGA	438
Описание алгоритма Madryga	439
Криптоанализ алгоритма Madryga	441
13.3. АЛГОРИТМ NEWDES	441
13.4. АЛГОРИТМ FEAL	443
Описание алгоритма FEAL	443
Криптоанализ алгоритма FEAL	446
Патенты	448
13.5. АЛГОРИТМ REDOC	448
Алгоритм REDOC III	449
Патенты и лицензии	450
13.6. АЛГОРИТМ LOKI	450
Алгоритм LOKI91	451
Описание алгоритма LOKI91	451
Криптоанализ алгоритма LOKI91	453
Патенты и лицензии	453
13.7. АЛГОРИТМЫ KHUFU и KHAFRE	454
Алгоритм Khufu	455
Алгоритм Khafre	455
Патенты	456
13.8. АЛГОРИТМ RC2	456
13.9. АЛГОРИТМ IDEA	458
Обзор алгоритма IDEA	459
Описание алгоритма IDEA	459
Скорость IDEA	462
Криптоанализ алгоритма IDEA	462
Режимы работы и варианты IDEA	465
Предостережение	466
Патенты и лицензии	466
13.10. АЛГОРИТМ MMB	466
Безопасность алгоритма MMB	468
13.11. АЛГОРИТМ CA-1.1	468
13.12. АЛГОРИТМ SKIPJACK	469

Глава 14. ДРУГИЕ БЛОЧНЫЕ ШИФРЫ	473
14.1. АЛГОРИТМ ГОСТ	473
Описание алгоритма ГОСТ	473
Криптоанализ алгоритма ГОСТ	476
14.2. АЛГОРИТМ CAST	477
14.3. АЛГОРИТМ BLOWFISH	479
Описание алгоритма Blowfish	479
Стойкость алгоритма Blowfish	482
14.4. АЛГОРИТМ SAFER	483
Описание алгоритма SAFER K-64	483
Алгоритм SAFER K-128	485
Стойкость алгоритма SAFER K-64	485
14.5. АЛГОРИТМ 3-WAY	486
Описание алгоритма 3-Way	486
14.6. АЛГОРИТМ CRAB	487
14.7. АЛГОРИТМ SXAL8/MBAL	489
14.8. АЛГОРИТМ RC5	489
14.9. ДРУГИЕ БЛОЧНЫЕ АЛГОРИТМЫ	491
14.10. ТЕОРИЯ ПРОЕКТИРОВАНИЯ БЛОЧНЫХ ШИФРОВ	492
Сети Фейстеля	493
Простые соотношения	493
Групповая структура	494
Слабые ключи	494
Устойчивость к дифференциальному и линейному криптоанализу	495
Проектирование S-блоков	495
Проектирование блочного шифра	498
14.11. ИСПОЛЬЗОВАНИЕ ОДНОСТОРОННИХ ХЕШ-ФУНКЦИЙ	499
Алгоритм Карна	499
Алгоритм Любы–Ракоффа	500
Шифр MDC	501
Безопасность шифров, основанных на односторонних хеш-функциях	502
14.12. ВЫБОР БЛОЧНОГО АЛГОРИТМА	503
Глава 15. КОМБИНИРОВАНИЕ БЛОЧНЫХ ШИФРОВ	505
15.1. ДВОЙНОЕ ШИФРОВАНИЕ	505
15.2. ТРОЙНОЕ ШИФРОВАНИЕ	507
Тройное шифрование с двумя ключами	507
Тройное шифрование с тремя ключами	509
Тройное шифрование с минимальным ключом (ТЕМК)	509

Режимы тройного шифрования	509
Варианты тройного шифрования	511
15.3. Удвоение длины блока	513
15.4. ДРУГИЕ СХЕМЫ МНОГОКРАТНОГО ШИФРОВАНИЯ	513
Двойной OFB/счетчик	514
Метод ECB + OFB	514
Схема xDES ⁱ	515
Пятикратное шифрование	517
15.5. Уменьшение длины ключа в алгоритме CDMF	517
15.6. ОТБЕЛИВАНИЕ	517
15.7. КАСКАДНОЕ ПРИМЕНЕНИЕ БЛОЧНЫХ АЛГОРИТМОВ	518
15.8. КОМБИНАЦИЯ НЕСКОЛЬКИХ БЛОЧНЫХ АЛГОРИТМОВ	519
Глава 16. ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ И ПОТОКОВЫЕ ШИФРЫ	521
16.1. ЛИНЕЙНЫЕ КОНГРУЭНТНЫЕ ГЕНЕРАТОРЫ	521
Комбинирование линейных конгруэнтных генераторов	523
16.2. РЕГИСТРЫ СДВИГА С ЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ	526
Программные реализации регистров LFSR	532
16.3. ПРОЕКТИРОВАНИЕ И АНАЛИЗ ПОТОКОВЫХ ШИФРОВ	534
Линейная сложность	534
Корреляционная стойкость	535
Другие атаки	536
16.4. ПОТОКОВЫЕ ШИФРЫ НА ОСНОВЕ РЕГИСТРОВ LFSR	536
Генератор Гейфе	537
Обобщенный генератор Гейфе	538
Генератор Дженнинга	538
Генератор “старт–стоп” Бета–Пайпера	539
Чередующийся генератор “старт–стоп”	540
Двусторонний генератор “старт–стоп”	541
Пороговый генератор	541
Самопрореживающие генераторы	542
Многоскоростной генератор скалярного произведения	542
Суммирующий генератор	544
Генератор DNRSG	544
Каскад Голлманна	544
Сжимающий генератор	545
Самосжимающий генератор	545
16.5. ШИФР A5	546
16.6. АЛГОРИТМ HUGHES XPD/KPD	547

16.7. АЛГОРИТМ NANOTEQ	548
16.8. АЛГОРИТМ RAMBUTAN	548
16.9. АДДИТИВНЫЕ ГЕНЕРАТОРЫ	549
Генератор Fish	549
Алгоритм Pike	550
Алгоритм Mush	550
16.10. АЛГОРИТМ ДЖИФФОРДА	551
16.11. АЛГОРИТМ M	552
16.12. АЛГОРИТМ PKZIP	553
Надежность алгоритма PKZIP	554
ГЛАВА 17. ДРУГИЕ ПОТОКОВЫЕ ШИФРЫ И ГЕНЕРАТОРЫ ИСТИННО СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ	555
17.1. АЛГОРИТМ RC4	555
17.2. АЛГОРИТМ SEAL	557
Семейство псевдослучайных функций	557
Описание алгоритма SEAL	558
Надежность алгоритма SEAL	559
Патенты и лицензии	560
17.3. АЛГОРИТМ WAKE	560
17.4. РЕГИСТРЫ СДВИГА С ОБРАТНОЙ СВЯЗЬЮ ПО ПЕРЕНОСУ	561
17.5. ПОТОКОВЫЕ ШИФРЫ НА ОСНОВЕ РЕГИСТРОВ FCSR	570
Каскадные генераторы	570
Комбинированные генераторы FCSR	570
Каскад LFSR/FCSR с суммированием/четностью	571
Чередующиеся генераторы “старт–стоп”	572
Сжимающие генераторы	573
17.6. РЕГИСТРЫ СДВИГА С НЕЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ	573
17.7. ДРУГИЕ ПОТОКОВЫЕ ШИФРЫ	575
Генератор Плесса	575
Генератор на основе клеточного автомата	576
Генератор $1/p$	576
Алгоритм $\text{sgupt}(1)$	576
Другие схемы	577
17.8. ПРОЕКТИРОВАНИЕ ПОТОКОВЫХ ШИФРОВ НА ОСНОВЕ ТЕОРИИ СИСТЕМ	577
17.9. ПРОЕКТИРОВАНИЕ ПОТОКОВЫХ ШИФРОВ НА ОСНОВЕ ТЕОРИИ СЛОЖНОСТИ	579
Генератор псевдослучайных чисел Шамира	579
Генератор Блюма–Микали	579

Генератор RSA	579
Генератор Блюма–Блюма–Шуба	580
17.10. ДРУГИЕ ПОДХОДЫ К ПРОЕКТИРОВАНИЮ ПОТОКОВЫХ ШИФРОВ	581
Шифр “Рип ван Винкль”	582
Рандомизированный потоковый шифр Диффи	582
Рандомизированный потоковый шифр Маурера	583
17.11. КАСКАД ИЗ НЕСКОЛЬКИХ ПОТОКОВЫХ ШИФРОВ	583
17.12. ВЫБОР ПОТОКОВОГО ШИФРА	584
17.13. ГЕНЕРИРОВАНИЕ НЕСКОЛЬКИХ ПОТОКОВ С ПОМОЩЬЮ ОДНОГО ГЕНЕРАТОРА ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ	584
17.14. ГЕНЕРАТОРЫ ИСТИННО СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ	586
Таблицы случайных чисел	586
Использование случайного шума	587
Использование таймера компьютера	589
Измерение задержек клавиатуры	590
Смещения и корреляции	590
Извлеченная случайность	591
ГЛАВА 18. ОДНОСТОРОННИЕ ХЕШ-ФУНКЦИИ	595
18.1. ОСНОВЫ	595
Длины односторонних хеш-функций	596
Обзор односторонних хеш-функций	597
18.2. АЛГОРИТМ SNEFRU	598
Криптоанализ алгоритма Snefru	599
18.3. АЛГОРИТМ N-ХЕШ	599
Криптоанализ алгоритма N-хеш	602
18.4. АЛГОРИТМ MD4	602
18.5. АЛГОРИТМ MD5	603
Описание алгоритма MD5	603
Стойкость MD5	608
18.6. АЛГОРИТМ MD2	608
18.7. АЛГОРИТМ SHA	609
Описание алгоритма SHA	610
Стойкость алгоритма SHA	613
18.8. АЛГОРИТМ RIPE-MD	614
18.9. АЛГОРИТМ HAVAL	614
18.10. ДРУГИЕ ОДНОСТОРОННИЕ ХЕШ-ФУНКЦИИ	615
18.11. ОДНОСТОРОННИЕ ХЕШ-ФУНКЦИИ НА ОСНОВЕ СИММЕТРИЧНЫХ БЛОЧНЫХ АЛГОРИТМОВ	616
Схемы, в которых длина хеш-значения равна длине блока	617

Модификация схемы Дэвиса–Майера	619
Схема Пренеля–Босселаерса–Говарца–Вандевалле	620
Алгоритм Кискатера–Жиро	620
Алгоритм LOKI с удвоенным блоком	621
Параллельная схема Дэвиса–Майера	621
Тандемная и синхронная схемы Дэвиса–Майера	621
Алгоритмы MDC-2 и MDC-4	623
Хеш-функция AR	624
Хеш-функция ГОСТ	625
Другие схемы	625
18.12. ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ С ОТКРЫТЫМ КЛЮЧОМ	626
18.13. ВЫБОР ОДНОСТОРОННЕЙ ХЕШ-ФУНКЦИИ	626
18.14. КОДЫ АУТЕНТИФИКАЦИИ СООБЩЕНИЙ	627
Алгоритм CBC-MAC	628
Алгоритм МАА	628
Двунаправленный алгоритм MAC	628
Методы Джунемана	629
Алгоритм RIPE-MAC	629
Алгоритм IBC-хеш	630
Односторонняя хеш-функция MAC	630
Алгоритм MAC с использованием потокового шифра	631
Глава 19. АЛГОРИТМЫ С ОТКРЫТЫМИ КЛЮЧАМИ	633
19.1. Основы	633
Стойкость алгоритмов с открытым ключом	634
19.2. АЛГОРИТМЫ НА ОСНОВЕ ЗАДАЧИ ОБ УКЛАДКЕ РАНЦА	634
Сверхвозрастающие ранцы	636
Создание открытого ключа из закрытого	637
Шифрование	637
Расшифровка	638
Практические реализации	638
Стойкость ранцевого метода	638
Варианты ранцевых алгоритмов	639
Патенты	639
19.3. АЛГОРИТМ RSA	640
Аппаратные реализации RSA	643
Скорость работы RSA	644
Программные ускорители	644
Стойкость алгоритма RSA	645
Атака с подобранным шифротекстом на RSA	646
Атака на RSA с использованием общего модуля RSA	647

Атака на RSA с использованием малого показателя шифрования	648
Атака на RSA с использованием малого показателя расшифровки	648
Выводы	649
Атака на шифрование и цифровую подпись с использованием алгоритма RSA	649
Стандарты	650
Патенты	650
19.4. СХЕМА ПОЛИГА—ХЕЛЛМАНА	650
Патенты	651
19.5. СХЕМА РАБИНА	651
Схема Уильямса	652
19.6. СХЕМА ЭЛЬ-ГАМАЛЯ	653
Подписи по схеме Эль-Гамала	653
Шифрование по схеме Эль-Гамала	655
Быстродействие	656
Патенты	656
19.7. СХЕМА МАКЭЛИСА	656
Другие алгоритмы, основанные на линейных кодах, исправляющих ошибки	657
19.8. КРИПТОСИСТЕМЫ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ	658
19.9. КРИПТОСИСТЕМА LUC	659
19.10. КРИПТОСИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ НА ОСНОВЕ КОНЕЧНЫХ АВТОМАТОВ	660
Глава 20. АЛГОРИТМЫ ЦИФРОВОЙ ПОДПИСИ С ОТКРЫТЫМ КЛЮЧОМ	663
20.1. АЛГОРИТМ ЦИФРОВОЙ ПОДПИСИ DSA	663
Реакция на заявление	664
Описание DSA	667
Ускоряющие предварительные вычисления	669
Генерация простых чисел DSA	670
Шифрование по схеме Эль-Гамала с алгоритмом DSA	671
Шифрование по алгоритму RSA с помощью алгоритма DSA	672
Стойкость алгоритма DSA	673
Атаки, направленные на параметр k	674
Опасности общего модуля	674
Скрытый канал в алгоритме DSA	675
Патенты	675
20.2. ВАРИАНТЫ АЛГОРИТМА DSA	676
20.3. АЛГОРИТМ ЦИФРОВОЙ ПОДПИСИ ГОСТ	678
20.4. СХЕМЫ ЦИФРОВОЙ ПОДПИСИ НА ОСНОВЕ ДИСКРЕТНЫХ ЛОГАРИФМОВ	679

20.5. СХЕМА ОНГА—ШНОРРА—ШАМИРА	681
20.6. СХЕМА ESIGN	682
Стойкость схемы ESIGN	683
Патенты	684
20.7. КЛЕТОЧНЫЕ АВТОМАТЫ	684
20.8. ДРУГИЕ АЛГОРИТМЫ С ОТКРЫТЫМ КЛЮЧОМ	684
ГЛАВА 21. СХЕМЫ ИДЕНТИФИКАЦИИ	687
21.1. СХЕМА ФЕЙГЕ—ФИАТА—ШАМИРА	687
Упрощенная схема идентификации Фейге—Фиата—Шамира	687
Схема идентификации Фейге—Фиата—Шамира	689
Пример	689
Улучшения протокола	691
Схема подписи Фиата—Шамира	691
Улучшенная схема подписи Фиата—Шамира	693
Другие улучшения	693
Схема идентификации Ота—Окамото	693
Патенты	693
21.2. СХЕМА ГИЛЛУ—КИСКАТЕ	693
Схема идентификации Гиллу—Кискате	694
Схема подписи Гиллу—Кискате	695
Несколько подписей	695
21.3. СХЕМА ШНОРРА	696
Протокол проверки подлинности	696
Протокол цифровой подписи	697
Патенты	698
21.4. ПРЕОБРАЗОВАНИЕ СХЕМ ИДЕНТИФИКАЦИИ В СХЕМЫ ПОДПИСИ	698
ГЛАВА 22. АЛГОРИТМЫ ОБМЕНА КЛЮЧАМИ	699
22.1. АЛГОРИТМ ДИФФИ—ХЕЛЛМАНА	699
Алгоритм Диффи—Хеллмана с тремя и более участниками	700
Расширенный алгоритм Диффи—Хеллмана	701
Алгоритм Хьюза	701
Обмен ключом без предварительного обмена данными	702
Патенты	702
22.2. ПРОТОКОЛ “СТАНЦИЯ—СТАНЦИЯ”	702
22.3. ТРЕХПРОХОДНЫЙ ПРОТОКОЛ ШАМИРА	703
22.4. ПРОТОКОЛ COMSET	705
22.5. ПРОТОКОЛ ОБМЕНА ЗАШИФРОВАННЫМИ КЛЮЧАМИ	705
Базовый протокол ЕКЕ	705

Реализация протокола ЕКЕ с помощью алгоритма RSA	706
Реализация протокола ЕКЕ с помощью схемы Эль-Гамала	707
Реализация протокола ЕКЕ с помощью алгоритма Диффи–Хеллмана	707
Усовершенствование протокола ЕКЕ	708
Расширенный протокол ЕКЕ	708
Применение протокола ЕКЕ	709
22.6. Защищенные переговоры о согласовании ключа	710
22.7. РАСПРЕДЕЛЕНИЕ КЛЮЧА ДЛЯ КОНФЕРЕНЦ-СВЯЗИ И СЕКРЕТНОЙ ШИРОКОВЕЩАТЕЛЬНОЙ ПЕРЕДАЧИ	711
Распределение ключей для конференции	713
Протокол Татебаяши–Мацузаки–Ньюмена	713
Глава 23. СПЕЦИАЛЬНЫЕ АЛГОРИТМЫ ДЛЯ ПРОТОКОЛОВ	715
23.1. Криптография с несколькими открытыми ключами	715
23.2. АЛГОРИТМЫ РАЗДЕЛЕНИЯ СЕКРЕТА	716
Схема интерполяционных многочленов Лагранжа	716
Векторная схема	717
Схема Асмута–Блума	718
Схема Карнина–Грини–Хеллмана	718
Более сложные пороговые схемы	718
Разделение секрета с мошенниками	719
23.3. СКРЫТЫЙ КАНАЛ	720
Скрытый канал на основе схемы Онга-Шнорра–Шамира	720
Скрытый канал на основе схемы Эль-Гамала	721
Скрытый канал на основе схемы ESIGN	722
Скрытый канал на основе схемы DSA	724
Уничтожение скрытого канала в схеме DSA	726
Другие схемы	727
23.4. НЕОСПОРИМЫЕ ЦИФРОВЫЕ ПОДПИСИ	727
Преобразуемые неоспоримые подписи	729
23.5. ПОДПИСИ, ПОДТВЕРЖДАЕМЫЕ ДОВЕРЕННЫМ ЛИЦОМ	730
23.6. ВЫЧИСЛЕНИЯ С ЗАШИФРОВАННЫМИ ДАННЫМИ	732
Задача дискретного логарифмирования	732
23.7. ЖЕРЕБЬЕВКА С ПОМОЩЬЮ ИДЕАЛЬНОЙ МОНЕТЫ	732
Жеребьевка с помощью идеальной монеты и квадратных корней	732
Жеребьевка с помощью идеальной монеты и возведения в степень по модулю p	733
Жеребьевка с помощью идеальной монеты и целых чисел Блума	734

23.8. Односторонние сумматоры	735
23.9. РАСКРЫТИЕ СЕКРЕТОВ ПО ПРИНЦИПУ “ВСЕ ИЛИ НИЧЕГО”	735
23.10. ЗАКОННЫЕ И ОТКАЗОУСТОЙЧИВЫЕ КРИПТОСИСТЕМЫ	739
Законная схема Диффи–Хеллмана	739
Отказоустойчивая схема Диффи–Хеллмана	740
23.11. ДОКАЗАТЕЛЬСТВО ЗНАНИЯ С НУЛЕВЫМ РАЗГЛАШЕНИЕМ	740
Доказательство знания дискретного логарифма с нулевым разглашением	740
Доказательство способности взломать алгоритм RSA с нулевым разглашением	741
Доказательство с нулевым разглашением того, что n является числом Блюма	742
23.12. СЛЕПЫЕ ПОДПИСИ	743
23.13. ЗАБЫВЧИВАЯ ПЕРЕДАЧА	743
23.14. ТАЙНЫЕ МНОГОСТОРОННИЕ ВЫЧИСЛЕНИЯ	744
Пример протокола	745
23.15. ВЕРОЯТНОСТНОЕ ШИФРОВАНИЕ	746
23.16. КВАНТОВАЯ КРИПТОГРАФИЯ	749

Часть IV

РЕАЛЬНЫЙ МИР 753

Глава 24. ПРИМЕРЫ РЕАЛИЗАЦИЙ	755
24.1. ПРОТОКОЛ КОМПАНИИ IBM для управления СЕКРЕТНЫМИ КЛЮЧАМИ	755
Модификация схемы	756
24.2. СИСТЕМА MITRENET	757
24.3. ТЕЛЕФОННЫЙ ТЕРМИНАЛ ISDN	758
Ключи	758
Вызов	759
24.4. STU-III	760
24.5. ПРОТОКОЛ KERBEROS	761
Модель Kerberos	761
Как работает Kerberos	762
Удостоверения	763
Сообщения Kerberos версии 5	764
Получение первоначального мандата	764
Получение серверных мандатов	765
Запрос к службе	766
Версия 4 протокола Kerberos	766
Стойкость протокола Kerberos	767

Лицензии	768
24.6. СИСТЕМА КРИПТОKNIGHT	768
24.7. СИСТЕМА SESAME	769
24.8. ОБЩАЯ КРИПТОГРАФИЧЕСКАЯ АРХИТЕКТУРА IBM	770
24.9. СХЕМА ПРОВЕРКИ ПОДЛИННОСТИ ISO	771
Сертификаты	772
Протоколы аутентификации	774
24.10. СТАНДАРТ PEM	776
Документы PEM	777
Сертификаты	778
Сообщения PEM	778
Безопасность стандарта PEM	781
Стандарт TIS/PEM	783
Программа RPEM	783
24.11. ПРОТОКОЛ БЕЗОПАСНОСТИ СООБЩЕНИЙ MSP	784
24.12. ПРОГРАММА PRETTY GOOD PRIVACY (PGP)	785
24.13. ИНТЕЛЛЕКТУАЛЬНЫЕ КАРТОЧКИ	788
24.14. СТАНДАРТЫ КРИПТОГРАФИИ С ОТКРЫТЫМ КЛЮЧОМ	789
24.15. УНИВЕРСАЛЬНАЯ СИСТЕМА ЭЛЕКТРОННЫХ ПЛАТЕЖЕЙ UEPS	792
24.16. МИКРОСХЕМА CLIPPER	794
24.17. МИКРОСХЕМА CAPSTONE	797
24.18. БЕЗОПАСНЫЙ ТЕЛЕФОН AT&T MODEL 3600 TELEPHONE SECURITY DEVICE (TSD)	798
Глава 25. ПОЛИТИЧЕСКИЕ ВОПРОСЫ	801
25.1. АГЕНТСТВО НАЦИОНАЛЬНОЙ БЕЗОПАСНОСТИ	801
Коммерческая программа CSEP	803
25.2. НАЦИОНАЛЬНЫЙ ЦЕНТР КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ (NCSC)	804
25.3. НАЦИОНАЛЬНЫЙ ИНСТИТУТ СТАНДАРТОВ И ТЕХНИКИ NIST	805
25.4. КОРПОРАЦИЯ RSA DATA SECURITY, INC.	809
25.5. КОРПОРАЦИЯ PUBLIC KEY PARTNERS	809
25.6. АССОЦИАЦИЯ IACR	811
25.7. КОНСОРЦИУМ RIPE	812
25.8. ПРОЕКТ SAFE	812
25.9. СТАНДАРТ ISO/IEC 9979	813
25.10. ПРОФЕССИОНАЛЬНЫЕ, ПРОМЫШЛЕННЫЕ И ПРАВООЗАЩИТНЫЕ ГРУППЫ	814
Центр EPIC	814
Фонд EFF	815
Ассоциация ACM	815

Институт IEEE	815
Ассоциация SPA	815
25.11. КОМПЬЮТЕРНАЯ СЕТЬ SCI.CRYPT	816
25.12. ШИФРОПАНКИ	816
25.13. ПАТЕНТЫ	817
25.14. ЭКСПОРТНОЕ ЗАКОНОДАТЕЛЬСТВО США	817
25.15. ЭКСПОРТ И ИМПОРТ КРИПТОГРАФИЧЕСКИХ СРЕДСТВ ЗА РУБЕЖОМ	826
25.16. ПРАВОВЫЕ ВОПРОСЫ	827
ПОСЛЕСЛОВИЕ МЭТТА БЛЕЙЗА	829

Часть V
ПРИЛОЖЕНИЕ 833

Исходные коды	835
DES	835
LOKI91	846
IDEA	853
ГОСТ	859
BLOWFISH	864
3-WAY	873
RC5	879
A5	883
SEAL	888
СПИСОК ЛИТЕРАТУРЫ	897
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	1021

Схемы идентификации

21.1. СХЕМА ФЕЙГЕ–ФИАТА–ШАМИРА

Схема цифровой подписи и аутентификации, разработанная Амосом Фиатом (Amos Fiat) и Ади Шамиром (Adi Shamir), обсуждается в [566, 567]. Уриель Фейге (Uriel Feige), Фиат и Шамир модифицировали алгоритм, превратив его в доказательство подлинности с нулевым знанием [544, 545]. Это лучшее доказательство подлинности с нулевым разглашением.

9 июля 1986 г. три автора подали заявку на получение патента США [1427]. Из-за потенциальных военных приложений заявка была рассмотрена оборонным ведомством. Иногда Патентное бюро выдает не патент, а документ, называемый приказом об установлении режима секретности. 6 января 1987 г., за три дня до истечения шестимесячного периода, по просьбе армии Патентное бюро издало такое распоряжение, заявив, что “...раскрытие или публикация предмета заявки ...может причинить ущерб национальной безопасности...”. Авторам было приказано уведомить всех граждан США, которые знали о проводимых исследованиях, что несанкционированное разглашение информации может привести к двум годам тюремного заключения, штрафу в 10000 долларов или тому и другому одновременно. Более того, авторы должны были сообщить комиссару по патентам и торговым знакам обо всех иностранных гражданах, которые получили доступ к этой информации.

Это было абсурдно. В течение второй половины 1986 г. авторы представляли свою работу на конференциях в Израиле, Европе и США. Они даже не были американскими гражданами, а вся работа была выполнена в Институте Вейцмана (Weizmann) в Израиле.

Информация стала распространяться в научном сообществе и прессе. В течение двух дней секретное распоряжение было аннулировано. Шамир и его коллеги считают, что на отмене секретного распоряжения настояло АНБ, хотя оно не давало никаких официальных комментариев. Дальнейшие подробности этой странной истории описаны в [936].

Упрощенная схема идентификации Фейге–Фиата–Шамира

Перед выдачей любых закрытых ключей арбитр выбирает случайный модуль, n , представляющий собой произведение двух больших простых чисел. В реальной жизни длина n должна быть не меньше 512 битов и как можно

ближе к 1024 битам. Модель n может общим для группы претендентов, доказывающих подлинность своих идентификационных данных. (Использование целых чисел Блума (Blum) упрощает вычисления, но не является необходимым для обеспечения безопасности.)

Для того чтобы сгенерировать открытый и закрытый ключи Пегги, доверенный арбитр выбирает число v , являющееся квадратичным вычетом по модулю n . Другими словами, выбирается число v , такое, что уравнение $x^2 \equiv v \pmod{n}$ имеет решение и существует $v^{-1} \pmod{n}$. Это число v и является открытым ключом Пегги. Затем вычисляется наименьшее число s , для которого выполняется условие $s \equiv \text{sqrt}(v^{-1}) \pmod{n}$. Это — закрытый ключ Пегги. Используется следующий протокол идентификации.

1. Пегги выбирает случайное число r , меньшее n . Затем она вычисляет число $x = r^2 \pmod{n}$ и посылает его Виктору.
2. Виктор посылает Пегги случайный бит b .
3. Если $b = 0$, то Пегги посылает Виктору число r . Если $b = 1$, то Пегги посылает Виктору число $y = r \times s \pmod{n}$.
4. Если $b = 0$, то Виктор проверяет условие $x = r^2 \pmod{n}$, убеждаясь, что Пегги знает значение $\text{sqrt}(x)$. Если $b = 1$, то Виктор проверяет условие $x = y^2 v \pmod{n}$, убеждаясь, что Пегги знает значение $\text{sqrt}(v^{-1})$.

Это единственный раунд протокола, называемый **аккредитацией**. Пегги и Виктор повторяют его t раз, пока Виктор не убедится, что Пегги знает число s . Протокол относится к категории “разделяй и выбирай”. Если Пегги не знает числа s , то она может подбирать разные числа r , чтобы обмануть Виктора, если он пошлет ей нуль или единицу. Она не может сделать одновременно и то и другое. Шансы на то, что ей удастся обмануть Виктора один раз, равны 50%. Вероятность, что ей удастся обмануть его t раз, равна $\frac{1}{2^t}$.

Виктор может атаковать протокол, выдавая себя за Пегги. Он может начать выполнение протокола с другим верификатором, Валерией. На шаге 1 вместо выбора случайного числа r ему останется просто использовать значение, которое Пегги использовала в прошлый раз. Однако вероятность того, что Валерия на шаге 2 выберет то же значение b , которое Виктор использовал в протоколе с Пегги, равна $\frac{1}{2}$. Следовательно, шансы на то, что он обманет Валерию, равны 50%. Вероятность, что ему удастся обмануть ее t раз, равна $\frac{1}{2^t}$.

Для того чтобы этот протокол работал, Пегги никогда не должна использовать число r повторно. В противном случае, если Виктор на шаге 2 пошлет

Пегги другой случайный бит, то получит оба ответа Пегги. Тогда даже по одному из них он сможет вычислить число s , и для Пегги все будет кончено.

Схема идентификации Фейге–Фиата–Шамира

В своих работах [544, 545] Фейге, Фиат и Шамир показали, как параллельная схема может повысить количество аккредитаций на раунд и уменьшить объем взаимодействия между Пегги и Виктором.

Сначала, как и в предыдущем примере, генерируется произведение двух больших простых чисел, n . Для генерации открытого и закрытого ключей Пегги сначала выбираются k разных чисел: v_1, v_2, \dots, v_k , где каждое v_i является квадратичным вычетом по модулю n . Иными словами, числа v_i выбираются так, чтобы уравнение $x^2 \equiv v_i \pmod{n}$ имело решение и существовало число $v_i^{-1} \pmod{n}$. Строка v_1, v_2, \dots, v_k является открытым ключом. Затем вычисляются наименьшие числа s_i , для которых $s_i = \text{sqrt}(v_i^{-1}) \pmod{n}$. Строка s_1, s_2, \dots, s_k является закрытым ключом.

Протокол имеет следующий вид.

1. Пегги выбирает случайное число r , меньшее n . Затем она вычисляет число $x = r^2 \pmod{n}$ и посылает его Виктору.
2. Виктор посылает Пегги строку из k случайных битов: b_1, b_2, \dots, b_k .
3. Пегги вычисляет число $y = r \left(s_1^{b_1} s_2^{b_2} \dots s_k^{b_k} \right) \pmod{n}$. (Она перемножает значения s_i , соответствующие $b_i = 1$. Если первым битом, посланным Виктором, будет единица, то число s_1 войдет в произведение, а если первым битом будет ноль, то нет, и т.д.) Пегги посылает число y Виктору.
4. Виктор проверяет условие $x = y^2 \left(v_1^{b_1} v_2^{b_2} \dots v_k^{b_k} \right) \pmod{n}$. (Он перемножает значения v_i , основываясь на случайной двоичной строке. Если его первым битом является единица, то число v_1 войдет в произведение, а если первым битом будет ноль, то нет, и т.д.)

Пегги и Виктор повторяют этот протокол t раз, пока Виктор не убедится, что Пегги знает числа s_1, s_2, \dots, s_k .

Вероятность, что Пегги удастся обмануть Виктора t раз, равна $\frac{1}{2^{kt}}$. Авторы рекомендуют использовать вероятность мошенничества $\frac{1}{2^{20}}$ и предлагают значения $k = 5$ и $t = 4$. Если вы страдаете паранойей, то увеличьте эти значения.

Пример

Рассмотрим работу этого протокола на небольших числах.

Если $n = 35$ (два простых множителя — 5 и 7), то возможными квадратичными вычетами являются числа:

- 1: $x^2 \equiv 1 \pmod{35}$ имеет решения: $x = 1, 6, 29, 34$.
 4: $x^2 \equiv 4 \pmod{35}$ имеет решения: $x = 2, 12, 23, 33$.
 9: $x^2 \equiv 9 \pmod{35}$ имеет решения: $x = 3, 17, 18, 32$.
 11: $x^2 \equiv 11 \pmod{35}$ имеет решения: $x = 9, 16, 19, 26$.
 14: $x^2 \equiv 14 \pmod{35}$ имеет решения: $x = 7, 28$.
 15: $x^2 \equiv 15 \pmod{35}$ имеет решения: $x = 15, 20$.
 16: $x^2 \equiv 16 \pmod{35}$ имеет решения: $x = 4, 11, 24, 31$.
 21: $x^2 \equiv 21 \pmod{35}$ имеет решения: $x = 14, 21$.
 25: $x^2 \equiv 25 \pmod{35}$ имеет решения: $x = 5, 30$.
 29: $x^2 \equiv 29 \pmod{35}$ имеет решения: $x = 8, 13, 22, 27$.
 30: $x^2 \equiv 30 \pmod{35}$ имеет решения: $x = 10, 25$.

Обратные значения по модулю 35 и их квадратные корни перечислены в следующей таблице.

v	v^{-1}	$x = \text{sqrt}(v^{-1})$
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Обратите внимание, что у чисел 14, 15, 21, 25 и 30 нет обратных значений по модулю 35, поскольку они не являются взаимно простыми с числом 35. Это имеет смысл, поскольку должно быть $(5-1)(7-1)/4$ квадратичных вычетов по модулю 35, взаимно простых с 35: $\text{НОД}(x, 35) = 1$ (см. раздел 11.3).

Итак, Пегги получает открытый ключ, состоящий из $k = 4$ значений: {4, 11, 16, 29}. Соответствующим закрытым ключом является {3, 4, 9, 8}. Рассмотрим один раунд протокола.

1. Пегги выбирает случайное число $r = 16$, вычисляет $16^2 \pmod{35} = 11$ и посылает его Виктору.
2. Виктор посылает Пегги строку случайных битов: {1, 1, 0, 1}
3. Пегги вычисляет число $16(3^1 \times 4^1 \times 9^0 \times 8^1) \pmod{35} = 31$ и посылает его Виктору.
4. Виктор проверяет условие $31^2(4^1 \times 11^1 \times 16^1 \times 29^1) \pmod{35} = 11$.

Пегги и Виктор повторяют этот протокол t раз, каждый раз с новым случайным числом r , пока Виктор не будет удовлетворен.

Небольшие числа, подобные использованным в примере, не обеспечивают реальной безопасности. Но если длина числа n равна 512 и более битам, то Виктор ничего не сможет узнать о закрытом ключе Пегги, кроме того, что Пегги действительно его знает.

Улучшения протокола

В протокол можно встроить идентификационные данные. Пусть I — это двоичная строка, представляющая идентификатор Пегги: имя, адрес, номер социального страхования, размер головного убора, любимый сорт прохладительного напитка и другая личная информация. Используем одностороннюю хеш-функцию $H(x)$ для вычисления $H(I, j)$, где j — небольшое число, добавленное к строке I . Найдем набор чисел j , для которых $H(I, j)$ представляет собой квадратичный вычет по модулю n . Эти значения $H(I, j)$ становятся строкой v_1, v_2, \dots, v_k (числа j не обязаны быть квадратичными вычетами). Теперь открытым ключом Пегги является строка I , и список чисел j . Пегги посылает строку I и список чисел j Виктору перед первым шагом протокола (в качестве альтернативы Виктор может загрузить эти значения с какой-нибудь открытой доски объявлений). Виктор генерирует строку v_1, v_2, \dots, v_k по значениям $H(I, j)$.

Теперь, после успешного завершения протокола, Виктор будет убежден, что Трент, которому известно разложение модуля на множители, сертифицировал связь между строкой I и Пегги, предоставив ей квадратные корни из v_i , полученные из строки I (см. раздел 5.2.) Фейге, Фиат и Шамир добавили следующие замечания [544, 545]:

Для неидеальных хеш-функций можно посоветовать рандомизировать строку I , добавляя к ней длинную случайную строку R . Эта строка выбирается арбитром и открывается Виктору вместе со строкой I .

В типичных реализациях число k должно быть от 1 до 18. Большие значения k могут уменьшить время и снизить трудности связи, уменьшая количество раундов.

Длина числа n должна быть не менее 512 битов. (Конечно, с тех пор в области факторизации достигнуты значительные успехи.)

Если каждый пользователь выберет свое собственное число n и опубликует его в файле открытых ключей, то можно обойтись и без арбитра. Однако такой RSA-подобный вариант делает схему гораздо менее удобной.

Схема подписи Фиата–Шамира

Превращение этой схемы идентификации в схему подписи — по существу, вопрос замены Виктора хеш-функцией. Главным преимуществом схемы цифровой подписи Фиата–Шамира над алгоритмом RSA является ее скорость:

для схемы Фиата–Шамира требуется всего лишь 1–4 процента модульных умножений, используемых в алгоритме RSA. В этом протоколе снова вернемся к Алисе и Бобу.

Используемые параметры не отличаются от параметров схемы идентификации. Выберем n — произведение двух больших простых чисел. Сгенерируем открытый ключ v_1, v_2, \dots, v_k и закрытый ключ s_1, s_2, \dots, s_k , где $s_i = \text{sqr}t(v_i^{-1}) \bmod n$.

1. Алиса выбирает t случайных целых чисел r_1, r_2, \dots, r_t в диапазоне от 1 до n и вычисляет числа x_1, x_2, \dots, x_t , такие, что $x_i = r_i^2 \bmod n$.
2. Алиса хеширует объединение сообщения и строки x_i , создавая поток битов: $H(m, x_1, x_2, \dots, x_t)$. Она использует первые kt битов этой строки в качестве значений b_{ij} , где i изменяется от 1 до t , j — от 1 до k .
3. Алиса вычисляет числа y_1, y_2, \dots, y_t , где

$$y_i = r_i \left(s_1^{b_{i1}} s_2^{b_{i2}} \dots s_k^{b_{ik}} \right) \bmod n.$$

(Для каждого i она перемножает значения s_j в зависимости от случайных значений b_{ij} . Если $b_{ij} = 1$, то s_j используется в вычислениях, если $b_{ij} = 0$, то нет.)

4. Алиса посылает Бобу число m , все биты b_{ij} и все значения y_i . У Боба уже есть открытый ключ Алисы, v_1, v_2, \dots, v_k .
5. Боб вычисляет числа z_1, z_2, \dots, z_t , где

$$z_i = y_i^2 \left(v_1^{b_{i1}} v_2^{b_{i2}} \dots v_k^{b_{ik}} \right) \bmod n.$$

(Как и прежде, Боб выполняет умножение в зависимости от значений b_{ij} .) Также обратите внимание на то, что z_i должно быть равно x_i .

6. Боб проверяет, что первые kt битов $H(m, z_1, z_2, \dots, z_t)$ — это значения b_{ij} , которые прислала ему Алиса.

Как и в схеме идентификации, безопасность схемы подписи пропорциональна $\frac{1}{2^{kt}}$. Она также зависит от сложности факторизации числа n . Фиат и Шамир показали, что подделка подписи упрощается, если сложность факторизации числа n значительно меньше 2^{kt} . Кроме того, из-за атаки на основе парадокса дней рождения (см. раздел 18.1) они рекомендуют повысить величину kt с 20 до (по крайней мере) 72, предлагая $k = 9$ и $t = 8$.

Улучшенная схема подписи Фиата–Шамира

Сильвия Микали (Silvia Micali) и Ади Шамир улучшили протокол Фиата–Шамира [1088]. Они выбирали числа v_1, v_2, \dots, v_k так, чтобы они были первыми k простыми числами:

$$v_1 = 2, v_2 = 3, v_3 = 5 \dots$$

Это — открытый ключ. Закрытым ключом, s_1, s_2, \dots, s_k , являются случайные квадратные корни, определяемые по формуле

$$s_i = \text{sqrt}(v_i^{-1}) \bmod n.$$

В этой версии у каждого участника должен быть свое число n . Такая модификация упрощает проверку подписей, не влияя на время генерации подписей и их стойкость.

Другие улучшения

На основе алгоритма Фиата–Шамира разработана N -сторонняя схема идентификации [264]. Два других улучшения схемы Фиата–Шамира описаны в [1218]. Еще один вариант изложен в [1368].

Схема идентификации Ота–Окамото

Этот протокол является вариантом схемы идентификации Фейге–Фиата–Шамира. Его стойкость основана на сложности факторизации целых чисел [1198, 1199]. Эти же авторы разработали схему с несколькими подписями (см. раздел 23.1), с помощью которой разные люди могут последовательно ставить цифровые подписи [1200]. Эта схема была предложена для реализации на интеллектуальных карточках [850].

Патенты

Схема Фиата–Шамира запатентована [1427]. Желающим получить лицензию на алгоритм необходимо обратиться по адресу Yeda Research and Development, The Weizmann Institute of Science, Rehovot 76100, Israel.

21.2. СХЕМА ГИЛЛУ–КИСКАТЕ

Схема Фейге–Фиата–Шамира была первым практическим протоколом идентификации. Этот протокол минимизировал вычисления, увеличивая количество итераций и аккредитаций на итерацию. Для ряда реализаций, например, для интеллектуальных карточек, это неприемлемо. Обмены информацией с внешним миром требуют времени, а хранение данных для каждой аккредитации может быстро исчерпать ограниченные возможности карточки.

Луи Гиллу (Louis Guillou) и Жан-Жак Кискате (Jean-Jacques Quisquater) разработали алгоритм идентификации с нулевым разглашением, который больше подходит для подобных приложений [670, 1280]. Обмены информацией между Пегги и Виктором, а также параллельные аккредитации в каждом обмене сведены к абсолютному минимуму: для каждого доказательства существует только один обмен, в котором предусмотрена только одна аккредитация. Для достижения того же уровня безопасности при использовании схемы Гиллу–Кискате потребуется выполнить в три раза больше вычислений, чем в схеме Фейге–Фиата–Шамира. Как и схему Фейге–Фиата–Шамира, этот алгоритм идентификации можно превратить в алгоритм цифровой подписи.

Схема идентификации Гиллу–Кискате

Предположим, что роль Пегги играет интеллектуальная карточка, которая собирается доказать свою подлинность Виктору. Идентификация Пегги проводится по ряду атрибутов, представляющих собой строку данных, содержащих название карточки, период ее действия, номер банковского счета и другие данные, подтверждающие ее правомочность. Эта битовая строка называется J . (В реальности строка атрибутов может быть очень длинной, а в качестве строки J используется ее хеш-значение. Такое усложнение никак не влияет на протокол.) Эта строка аналогична открытому ключу. Другой открытой информацией, общей для всех “Пегги”, которые могут использовать это приложение, является показатель степени v и модуль n , где n — произведение двух секретных простых чисел. Закрытым ключом служит число B , удовлетворяющее условию $JB^v \equiv 1 \pmod{n}$.

Пегги посылает Виктору свои атрибуты J . Теперь она хочет доказать Виктору, что это именно ее атрибуты. Для этого она должна убедить Виктора, что ей известно число B . Вот как выглядит этот протокол.

1. Пегги выбирает случайное целое r , лежащее в диапазоне от 1 до $n-1$. Она вычисляет число $T = r^v \pmod{n}$ и отправляет его Виктору.
2. Виктор выбирает случайное целое d , находящееся в диапазоне от 0 до $v-1$, и посылает число d Пегги.
3. Пегги вычисляет $D = rB^d \pmod{n}$ и посылает его Виктору.
4. Виктор вычисляет число $T' = D^v J^d \pmod{n}$. Если $T \equiv T' \pmod{n}$, то подлинность Пегги доказана.

Математическое доказательство этого протокола не слишком сложное:

$$T' = D^v J^d = (rB^d)^v J^d = r^v B^{dv} J^d = r^v (B^v J)^d = r^v = r' \equiv T \pmod{n},$$

так как $JB^v \equiv 1 \pmod{n}$.

Схема подписи Гиллу–Кискате

Эту схему идентификации можно преобразовать в схему подписи, также пригодную для реализации в интеллектуальных карточках [671, 672]. Открытый и закрытый ключи не меняются. Вот как выглядит протокол.

1. Алиса выбирает случайное целое r , находящееся в диапазоне от 1 до $n-1$. Она вычисляет $T = r^v \bmod n$.
2. Алиса вычисляет число $d = H(M, T)$, где M — подписываемое сообщение, а $H(x)$ — односторонняя хеш-функция. Значение d , полученное с помощью хеш-функции, должно лежать в диапазоне от 0 до $v-1$ [1280]. Если значение хеш-функции выходит за пределы этого диапазона, то оно должно быть приведено по модулю v .
3. Алиса вычисляет число $D = rB^d \bmod n$. Подпись состоит из сообщения M , двух вычисленных значений, d и D , и ее атрибутов J . Алиса посылает подпись Бобу.
4. Боб вычисляет $T' = D^v J^d \bmod n$. Затем он вычисляет $d' = H(M, T')$. Если $d = d'$, то Алиса знает число B и ее подпись является действительной.

Несколько подписей

Что произойдет, если несколько человек захотят подписать один и тот же документ? Проще всего, чтобы они подписали его по-отдельности, но рассматриваемая схема подписи делает это лучше. Пусть Алиса и Боб подписывают документ, а Кэрол проверяет подписи, но в процесс подписания может быть вовлечено произвольное количество людей. Как и раньше, Алиса и Боб обладают уникальными значениями J и B : (J_A, B_A) и (J_B, B_B) . Значения n и v являются общими для всей системы.

1. Алиса выбирает случайное целое r_A , находящееся в диапазоне от 1 до $n-1$. Она вычисляет число $T_A = r_A^v \bmod n$ и посылает число T_A Бобу.
2. Боб выбирает случайное целое r_B , находящееся в диапазоне от 1 до $n-1$. Он вычисляет число $T_B = r_B^v \bmod n$ и посылает число T_B Алисе.
3. Алиса и Боб вычисляют число $T = (T_A \times T_B) \bmod n$.
4. И Алиса, и Боб вычисляют $d = H(M, T)$, где M — подписываемое сообщение, а $H(x)$ — односторонняя хеш-функция. Значение d , полученное с помощью хеш-функции, должно лежать в диапазоне от 0 до $v-1$ [1280]. Если значение хеш-функции выходит за пределы этого диапазона, то он должен быть приведен по модулю v .
5. Алиса вычисляет $D_A = r_A B_A^d \bmod n$ и посылает D_A Бобу.

6. Боб вычисляет $D_B = r_B B_B^d \bmod n$ и посылает D_B Алисе.
7. И Алиса, и Боб вычисляют $D = D_A D_B \bmod n$. Подпись состоит из сообщения M , двух вычисленных значений, d и D , и атрибутов обоих подписывающих: J_A и J_B .
8. Кэрл вычисляет $J = J_A J_B \bmod n$.
9. Кэрл вычисляет $T' = D^v J^d \bmod n$. Затем она вычисляет $d' = H(M, T')$. Если $d \equiv d'$, то множественная подпись является действительной.

Этот протокол может быть расширен на произвольное количество людей. Для этого люди, подписывающие сообщение, должны перемножить свои значения T_i на шаге 3 и свои значения D_i на шаге 7. Для того чтобы проверить множественную подпись, необходимо на шаге 8 перемножить значения всех подписывающих J_i . Либо все подписи правильны, либо существует по крайней мере одна неправильная подпись.

21.3. СХЕМА ШНОРРА

Стойкость схемы проверки подлинности и подписи, разработанная Клаусом Шнорром [1396,1397], опирается на трудность вычисления дискретных логарифмов. Для генерации пары ключей сначала выбираются два простых числа, p и q , так, чтобы число q было множителем числа $p-1$. Затем выбирается число a , не равное единице, такое, что $a^q \equiv 1 \pmod{p}$. Все эти числа можно свободно опубликовать и использовать в группе пользователей.

Для генерации конкретной пары ключей выбирается случайное число, меньшее q . Оно является закрытым ключом, s . Затем вычисляется открытый ключ $v = a^{-s} \bmod p$.

Протокол проверки подлинности

1. Пегги выбирает случайное число r , меньшее q , и вычисляет $x = a^r \bmod p$. Эти вычисления являются предварительными и могут быть выполнены задолго до появления Виктора.
2. Пегги посылает Виктору число x .
3. Виктор посылает Пегги случайное число e из диапазона $0 - 2^t - 1$. (Что такое t , я объясню позднее.)
4. Пегги вычисляет число $y = (r + se) \bmod q$ и посылает его Виктору.
5. Виктор проверяет условие $x = a^y v^e \bmod p$.

Стойкость алгоритма зависит от параметра t . Сложность взлома алгоритма равна примерно 2^t . Шнорр советует использовать число p длиной около 512 битов, q — около 140 битов и t — 72 бита.

Протокол цифровой подписи

Алгоритм Шнорра можно использовать и в качестве протокола цифровой подписи сообщения M . В этом случае используется та же самая пара ключей, но добавляется односторонняя хеш-функция $H(M)$.

1. Алиса выбирает случайное число r , меньшее q , и вычисляет $x = a^r \bmod p$. Это стадия предварительных вычислений.
2. Алиса объединяет M и x и хеширует результат:

$$e = H(M, x).$$

3. Алиса вычисляет число $y = (r + se) \bmod q$. Подписью являются значения e и y , которые она посылает Бобу.
4. Боб вычисляет число $x' = a^y v^e \bmod p$. Затем он проверяет, что хеш-значение для конкатенации M и x' равно e :

$$e = H(M, x').$$

5. Если это так, то он считает подпись верной.

В своей работе Шнорр приводит следующие новые свойства своего алгоритма:

Большая часть вычислений, необходимых для генерации подписи и не зависящих от подписываемого сообщения, может быть выполнена на стадии предварительных вычислений. Следовательно, эти вычисления можно выполнить во время простоя и они не влияют на скорость подписания. Атака, направленная на стадию предварительных вычислений, рассматривается в [475], но я не думаю, что она имеет практическую ценность.

При одинаковом уровне безопасности длина подписей в схеме Шнорра короче, чем в алгоритме RSA. Например, при 140-битовом числе q длина подписей равна всего лишь 212 битам, т.е. меньше половины длины подписей RSA. Подписи по схеме Шнорра также намного короче подписей по схеме Эль-Гамала.

Конечно, из практических соображений количество битов, используемых в этой схеме, может быть уменьшено: например, для схемы идентификации, в которой мошенник должен выполнить атаку в онлайн-режиме всего за несколько секунд (сравните со схемой подписи, когда мошенник может годами вести расчеты, чтобы выполнить подлог).

Модификация, выполненная Эрни Брикеллом (Ernie Brickell) и Кевином МакКерли (Kevin McCurley), повысила стойкость этого алгоритма [265].

Патенты

Схема Шнорра запатентована в США [1398] и во многих других странах. В 1993 г. компания РКР приобрела общемировые права на этот патент (см. раздел 25.5). Срок действия патента США истекает 19 февраля 2008 г.

21.4. ПРЕОБРАЗОВАНИЕ СХЕМ ИДЕНТИФИКАЦИИ В СХЕМЫ ПОДПИСИ

Стандартный метод преобразования схемы идентификации в схему подписи — замена Виктора односторонней хеш-функцией. Перед подписанием сообщение не хешируется, вместо этого хеширование встраивается в алгоритм подписи. В принципе, такую манипуляцию можно проделать с любой схемой идентификации.