Об одном способе формализации свойства анонимной аутентификации

А. О. Бахарев, В. С. Бельский, И. Ю. Герасимов, К. Д. Царегородцев

***Аннотация*—В работе рассматривается подход к формализации свойства анонимной аутентификации в рамках парадигмы «доказуемая стойкость». Описан псевдокод модели, приведены комментарии относительно возможностей противника, охватываемых моделью, описываемых моделью свойств безопасности. Рассмотрено множество атак, формализуемых в рамках модели, приведено сравнение модели с аналогичными моделями для (P)AKE-протоколов, указаны возможности для расширения модели. Приводится пошаговое описание примера формализации для одного конкретного протокола аутентификации, основанного на процедуре аутентификации и выработки общего ключа в сетях 5G.**

***Ключевые слова*—идентификация, аутентификация, анонимность, приватность, доказуемая стойкость**

I. Введение

В настоящей работе описывается формальная модель анонимной аутентификации σAuth в рамках подхода «доказуемая стойкость» (теоретико-сложностные сведения). Модель σAuth задается в соответствии с основными принципами построения моделей для AKE-протоколов [1, 2, 3, 4, 5, 6, 7], аутентификации [8, 9] и анонимности участников [10, 11, 12, 13, 14, 15]. Свойства анонимности и безопасности аутентификации в модели σAuth формализуются в виде угрозы различимости: противнику необходимо по поведению оракулов внутри эксперимента определить, каким именно значением был зафиксирован бит b, задающий поведение взаимосвязанных оракулов. Для полной спецификации модели необходимо определить ряд «гиперпараметров» модели: набор предикатов, которые задаются в соответствии с предъявляемыми к протоколу свойствами безопасности. Работа построена следующим образом:

- в разделе II задается собственно модель σAuth, перед описанием псевдокода эксперимента даются вводные замечания касательно требуемых свойств безопасности и возможностей противника, приводится формальная модель интерактивного протокола, рассматриваются понятия сеанса и внутреннего состояния,
- раздел III посвящен обсуждению модели противника: как именно формализуются требования безопасности, какие возможности противника включены в модель, каковы её ограничения и какие расширения модели можно предложить для тех или иных ситуаций,
- в разделе IV рассматривается применение модели для формализации свойств безопасности одного конкретного протокола аутентификации,
- в разделе V описаны атаки на указанный протокол в рамках модели σAuth.

Модель σAuth может быть использована для доказательства стойкости различных протоколов анонимной аутентификации.

Статья получена 3 марта 2025.

Бахарев Александр Олегович, Лаборатория Криптографии АО «НПК «Криптонит», (email: a.bacharev@kryptonite.ru).

Бельский Владимир Сергеевич, Лаборатория Криптографии АО «НПК «Криптонит», (email: v.belsky@kryptonite.ru).

Герасимов Илья Юрьевич, Лаборатория Криптографии АО «НПК «Криптонит», (email: i.gerasimov@kryptonite.ru).

Царегородцев Кирилл Денисович, Лаборатория Криптографии АО «НПК «Криптонит», (email: kirill94_12@mail.ru).

A. Используемые обозначения

Под $x \leftarrow y$ мы подразумеваем присвоение значения y переменной x; если x и y — структуры данных, то под $x \leftarrow y$ подразумевается присвоение $x.a \leftarrow y.a$ для всех общих для структур x и y полей a. Запись $x \xleftarrow{\$} \mathcal{O}$ символизирует процесс запуска вероятностного алгоритма $\mathcal{O}$ с присвоением результата работы алгоритма переменной x, при этом распределение неявно задается алгоритмом $\mathcal{O}$; через $x \xleftarrow{\mathcal{U}} M$, где M — конечное множество, мы обозначаем операцию присвоения переменной x значения, выбранного в соответствии с равномерным распределением на M. Пустой словарь обозначается через $[\,]$. Конкатенация строк x и y обозначается через $x \,\|\, y$. Символ $\perp$ означает либо символ ошибки (например, ошибка расшифрования), либо специальный символ, означающий отсутствие элемента в словаре по заданному индексу. Если X — некоторый массив, индексируемый ключами (s, t), то через $\cup_t X[s, t]$ будем обозначать словарь значений, который индексу t сопоставляет значение $X[s, t]$ для всех t, при которых $X[s, t] \neq \perp$. Мы будем использовать символ ∞ и предполагать, что $n < \infty$ для любого натурального n.

II. Модель безопасности σAuth

В настоящем разделе рассмотрим модель безопасности σAuth для интерактивного анонимного протокола аутентификации (см. также [1, 8, 16, 17]). Рассматриваемая ниже модель формализует часть взаимодействия между аутентифицирующимся участником (далее — абонентом) и выделенной доверенной стороной (далее — ДС) и задает свойства анонимности абонентов и безопасности аутентификации (где анонимность понимается в смысле анонимности от третьих лиц) в «квантифицируемой форме» (с помощью преимущества противника, подробнее см., например, [2, 18]). При этом из рассмотрения исключаются возможности противника по влиянию на

предварительный этап (инициализацию) ДС и абонентов. Модель учитывает следующие свойства безопасности:

- **явная аутентификация участников в ходе работы протокола:** противник не может успешно завершить протокол аутентификации без содействия легитимного участника;
- **анонимность:** противник не получает никакой информации о том, какие именно абоненты взаимодействуют с ДС в различных сеансах протокола (в том числе при компрометации долговременного внутреннего состояния абонента).

A. Возможности противника

Мы будем предполагать, что канал между абонентом и ДС является незащищенным. В таких условиях для описания среды, в которой функционирует протокол, используется модель Долева-Яо (см., например, [19, раздел 2.3]). Согласно модели, в уязвимой среде злоумышленник обладает следующими возможностями:

- получать любое сообщение, передаваемое по сети;
- устанавливать соединение с любым другим абонентом от имени любого другого абонента;
- перехватывать, задерживать, перемешивать передаваемые сообщения.

Также противник может обладать дополнительными возможностями, связанными с регистрацией абонентов, навязыванием и компрометацией ключей или промежуточных (эфемерных) значений в протоколе и т.д. (более подробно априорные возможности противника по взаимодействию с системой описаны, например, в [20]).

Мы будем рассматривать противников в некоторой фиксированной модели вычислений (например, машины Тьюринга). В таком случае под временными ресурсами противника $\mathcal{A}$ мы будем понимать величину t, ограничивающую как время работы противника (число тактов вычислений), так и размер программы (кода) противника.

Модель безопасности далее в тексте описывается посредством т.н. эксперимента, формализующего некоторое интуитивно понятное свойство безопасности в рамках парадигмы «доказуемая стойкость» (см., например, [2, 21, 22]). Для этого в эксперименте описывается набор оракулов (интерфейсов), доступных противнику, а также задается мера успеха противника (преимущество). Противник может делать адаптивные запросы к оракулам и получать от них ответы, которые могут содержать какую-либо информацию о секретных (неизвестных противнику) значениях (например, информацию о ключах схемы шифрования). Задачей противника является либо подделка какого-либо значения в ходе эксперимента, либо различение поведения оракулов в двух различных экспериментах.

B. Сеансы и внутренние состояния в интерактивных протоколах

Под протоколом будем подразумевать [17] распределенный алгоритм, в процессе выполнения которого участники последовательно выполняют определенные действия и обмениваются сообщениями. В [17] предлагается следующее определение сеанса протокола: конкретная реализация протокола с конкретными участниками.

Пусть Π — некоторый интерактивный протокол между двумя участниками A и B.

Определение 1. *Под **внутренним состоянием** участника в протоколе Π будем понимать пару $st = (ltp, stp)$, состоящую из:*

- *долговременных параметров (long-term parameters) участника протокола ltp, например, долговременного симметричного ключа участника, открытого ключа ДС, счетчика числа соединений, аутентификационных данных абонента и т.д.),*
- *кратковременных (эфемерных) параметров (short-term parameters) участника протокола, например, номер этапа протокола, на котором находится участник, а также все принятые до настоящего момента сообщения в ходе работы протокола (стенограмма)).*

Замечание 1. *В настоящей работе под «эфемерными» мы понимаем те параметры, которые привязаны к конкретному сеансу работы протокола, под «долговременными» — те параметры, которые сохраняются между сеансами работы протокола: счетчики, ключи и т.д. Разделение на долговременные и эфемерные параметры связано с тем, что часть параметров связана только с конкретным сеансом протокола (например шаг работы протокола или стенограмма сеанса), а часть сохраняется между различными сеансами (например, ключи). Таким образом, долговременные параметры привязаны к абонентам, а кратковременные — к сеансам.*

Для дальнейших рассмотрений под **сеансом** протокола будем понимать следующую тройку:

- держатель сеанса — идентификатора участника протокола,
- партнер — идентификатора участника протокола,
- внутреннее состояние держателя сеанса.

Держатель и партнер соответствуют «конкретным участникам» сеанса протокола, а внутреннее состояние держателя определяет «конкретную реализацию» протокола.

Замечание 2. *В рамках работ [2, 8] сеансы протокола задаются с помощью оракулов-участников, которые в неявном виде хранят всю необходимую информацию о сеансе протокола (как эфемерные, так и долговременные параметры).*

Понятие держателя сеанса не совпадает с понятием инициатора соединения (Сервера в терминологии протокола TLS). Сеансы и их держатели являются «техническими артефактами» модели и не должны соответствовать никаким конкретным физическим характеристикам системы (также см. [2, раздел 1.2], где отмечается, что сеансам можно поставить в соответствие конкретные процессы в рамках ОС, реализующие протокольное взаимодействие пользователя). Эти понятия вводятся для определения класса нетривиальных атак на интерактивный протокол: так, мы хотим иметь возможность исключить из рассмотрения «атаки», при которых противник просто пересылает сообщения от одного участника к другому. Для этого нам необходимо вводить понятие сеансов (в частности, сопряженных сеансов, см. замечание 9) и держателей/партнеров.

Понятие сеанса и внутреннего состояния в сеансе соответствуют аналогичным понятиям, используемым для анализа других интерактивных протоколов. Так, например, для протокола TLS в модели из работы [1] вводятся понятия держателя сеанса (session owner, понятие аналогично рассматриваемому в настоящей работе); метки сеанса (session label, соответствует указателю π в модели ниже); состояния сеанса (соответствует параметру res в рассматриваемой ниже модели). В работе [2] держатель сеанса определяется с помощью оракула-участника (соответствует конкретному держателю), метка сеанса i в указанной работе является локальным (а не глобальным) параметром оракула-участника, состояние сеанса задается параметром acc.

C. Протокол анонимной аутентификации

Приведем определение для основного объекта изучения — протокола анонимной аутентификации.

Определение 2. *Протокол (схема) анонимной аутентификации с выделенной доверенной стороной — тройка (вероятностных) алгоритмов*

$$\Pi = (\mathsf{InitTP}, \mathsf{InitUser}, \mathsf{Auth})$$

1) $\Pi.\mathsf{InitTP}$*: алгоритм инициализации выделенного участника — ДС, возвращающий структуру TP — начальные долговременные параметры ДС.*
2) $\Pi.\mathsf{InitUser}(ID, TP)$*: алгоритм инициализации абонента, на вход принимающий уникальный идентификатор абонента ID и долговременные параметры ДС TP и возвращающий долговременные параметры для связи абонента и ДС и долговременные параметры для связи ДС и абонента.*
3) $\Pi.\mathsf{Auth}(ltp, stp, m)$*: алгоритм анонимной аутентификации, на вход принимающий тройку:*
   - (ltp, stp) *— внутреннее состояние участника в сеансе,*
   - *текущее сообщение m,*

   и возвращающий тройку:
   - (ltp', stp') *— обновленное внутреннее состояние участника в сеансе,*
   - *сообщение-ответ m'.*

В рассматриваемой ниже модели для протокола аутентификации структура эфемерных параметров stp состоит из следующих полей:

- $stp.step$: текущий шаг работы, задаваемый в соответствии со спецификацией протокола;
- $stp.res$: результат сеанса — в процессе (in-progress), успешная аутентификация (accept) или ошибка ($\perp$);
- $stp.trans$: стенограмма сеанса связи (до настоящего момента);
- $stp.holder$: держатель сеанса;
- $stp.par$: предполагаемый партнер в сеансе связи (для абонента партнером всегда является $TrPar$ (Trusted Party, ДС), для ДС — один из абонентов ID);
- $stp.paired$: пара идентификаторов, ассоциированных с виртуальным идентификатором абонента vid, для которого был запущен сеанс во время вызова оракула UserSession;
- $stp.start$: «время» начала сеанса (относительно других сеансов);
- $stp.finish$: «время» окончания сеанса (относительно других сеансов).

Будем предполагать, что участники протокола начинают сеанс протокола с шага init (начальное состояние) и завершают на шаге fin (успешное завершение сеанса аутентификации или ошибка в ходе сеанса). Также в ходе работы протокола участники могут находиться на некоторых промежуточных шагах.

Замечание 3. *Иногда для учета более сложных атак для поля $stp.res$ необходимо рассматривать несколько типов различимых ошибок (см., например, [23], а также атаки [24, 25]).*

Замечание 4. *При рассмотрении протоколов аутентификации с равноправными участниками предполагаемым партнером может быть любой другой абонент, не совпадающий с держателем. Также в интерактивных протоколах более общего вида во внутреннем состоянии в качестве эфемерных параметров могут выделяться дополнительные поля: выработанный сеансовый ключ, роль участника в сеансе (инициатор/ответчик, см., например, [26]) и т.д.).*

D. Модель безопасности $\sigma\mathsf{Auth}$

В рамках используемого далее подхода «доказуемая стойкость» противник взаимодействует с окружением (Экспериментатором) посредством набора оракулов, которые формализуют возможности противника по взаимодействию с реальной системой. Экспериментатор ведет следующие согласованные словари.

Словарь долговременных параметров $LTP[\mathsf{A}, \mathsf{B}]$, элементами словаря являются долговременные параметры участника A для взаимодействия с фиксированным участником B. Словарь заполняется в ходе вызовов оракула $\mathsf{CreateUser}(ID)$, элементы словаря изменяются в ходе работы протокола и при запуске нового сеанса связи.

Словарь кратковременных параметров $STP[\pi]$, элементами словаря являются кратковременные параметры держателя сеанса π в сеансе π.

Замечание 5. *Мы будем считать, что если не оговорено иное, то значениями по умолчанию (default) для кратковременных параметров сеанса являются следующие:*

$$STP[\pi].step \leftarrow \mathsf{init},\ STP[\pi].res \leftarrow \mathsf{in\text{-}progress},$$
$$STP[\pi].holder \leftarrow \perp,\ STP[\pi].par \leftarrow \perp,$$
$$STP[\pi].paired \leftarrow \perp, STP[\pi].trans \leftarrow \epsilon,$$
$$STP[\pi].start \leftarrow \pi,\ STP[\pi].finish \leftarrow \infty$$

Мы будем писать $STP[\pi] \leftarrow default$ для операции инициализации структуры сеанса с помощью указанных полей по умолчанию.

Замечание 6. *Название «время» для полей $start$ и $finish$ в структуре сеансов несколько условно: в рассматриваемой ниже модели временем считается количество когда-либо открытых к настоящему моменту сеансов. Так, например, сеанс π получит время начала сеанса*

$$STP[\pi].start \leftarrow \pi.$$

При этом время окончания сеанса π будет зависеть от того, сколько сеансов будет открыто после открытия

сеанса π: если с момента открытия сеанса π до момента его закрытия было открыто $k \geq 0$ других сеансов, то время закрытия будет равным

$$STP[\pi].finish \leftarrow \pi + k.$$

В частности, «времена» начала и конца сеанса π удовлетворяют неравенству

$$STP[\pi].start \leq STP[\pi].finish.$$

Словарь виртуальных идентификаторов ID^{virt}, необходимый для формализации понятия анонимности абонентов (см. ниже). Также в ходе эксперимента для формализации понятия анонимности используются множества ID^{free}, ID^{drawn}.

Структура ДС TP; структура TP изменяется при запуске нового сеанса связи и в ходе работы протокола.

Множество скомпрометированных абонентов $Corr$, элементами которого являются пары, состоящие из идентификатора взломанного абонента ID и «времени» компрометации π.

Задачей противника в рамках эксперимента является корректное определение секретного бита b, от которого зависят распределения ответов оракулов. Рассмотрение релевантности модели, ее возможности и ограничения, приведено в разделе III.

Определение 3. *Преобладание противника $\mathcal{A}$ в модели σAuth с набором предикатов*

$$(\mathsf{Match}, \mathsf{Rand}, \mathsf{Corruption})$$

для протокола анонимной аутентификации $\Pi = (\mathsf{InitTP}, \mathsf{InitUser}, \mathsf{Auth})$ *задается равенством*

$$\mathbf{Adv}_{\Pi}^{\sigma\mathsf{Auth}}(\mathcal{A}) = \\ = \Pr\left[\mathbf{Exp}_{\Pi}^{\sigma\mathsf{Auth\text{-}1}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\Pi}^{\sigma\mathsf{Auth\text{-}0}}(\mathcal{A}) \to 1\right],$$

псевдокод эксперимента $\mathbf{Exp}_{\Pi}^{\sigma\mathsf{Auth}\text{-}b}$, $b \in \{0,1\}$, *приведен на рис. 1. Через $\mathcal{O}$ обозначен набор оракулов* CreateUser, TPSession, UserSession, Send, Free, Result, Corrupt. *Псевдокод используемых противником оракулов приведен на рис. 2.*

$\mathbf{Exp}_{\Pi}^{\sigma\mathsf{Auth}\text{-}b}(\mathcal{A})$

$\pi \leftarrow 0$
$LTP \leftarrow [\,]$
$STP \leftarrow [\,]$
$Corr \leftarrow \emptyset$
$vid \leftarrow 0$
$ID^{free} \leftarrow \emptyset$
$ID^{drawn} \leftarrow \emptyset$
$ID^{virt} \leftarrow [\,]$
$TP \leftarrow \Pi.\mathsf{InitTP}()$
$LTP[TrPar, \perp] \leftarrow TP$
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O},\mathsf{Draw}^b,\mathsf{TestAuth}^b}(TP.pk)$
return b'

Рис. 1: Псевдокод эксперимента σAuth для протокола Π

Определение 4. *Обозначим через*

$$\mathbf{InSec}_{\Pi}^{\sigma\mathsf{Auth}}(t, P, Q, \Theta, r, d)$$

*максимум среди преимуществ противников $\mathcal{A}$ в модели σ*Auth *для протокола* Π, *где противник $\mathcal{A}$ имеет ограничение t на вычислительные ресурсы и следующие ограничения на обращения к оракулам:*

- *число запросов к оракулу* CreateUser *(число различных абонентов) не превышает d,*
- *число компрометаций абонентов (число обращений к оракулу* Corrupt*) не превышает r,*
- *число запущенных сеансов с идентификатором vid, который был получен при связывании ID_i с некоторым ID', не превышает $P[i]$, $1 \leq i \leq d$,*
- *число запущенных сеансов со стороны ДС, предполагаемый партнер в которых равен ID_i, не превышает $Q[i]$, $1 \leq i \leq d$,*
- *число запросов к оракулу* Draw^b *(максимальное число связываний) с идентификатором ID_i не превышает $\Theta[i]$, $1 \leq i \leq d$.*

*E. Предикаты модели σ*Auth

Предикаты Match, Rand, Corruption задаются в соответствии с рассматриваемым конкретным протоколом анонимной аутентификации (см. раздел IV с примером задания предикатов для конкретного протокола). Фактически, указанные предикаты являются «гиперпараметрами» рассматриваемой модели и «настраиваются» в зависимости от рассматриваемого протокола и требуемых от него свойств безопасности.

Предикат Match формализует понятие «нетривиальной атаки». Тривиальной можно считать такую атаку, в ходе которой противник просто пересылает сообщения из одного сеанса в другой без какого-либо изменения (см., однако, работы [27, 28, 29], рассматривающие такие протоколы аутентификации, для которых простая пересылка сообщений может приводить к угрозам безопасности и не должна исключаться как «тривиальная»). Также если протокол не обеспечивает свойств безопасности при компрометации участника (защиту от «чтения назад» [30], анонимность при «чтении назад» [10]), то «тривиальной» может считаться атака, в ходе которой противник сначала провел взаимодействие с некоторым абонентом, а затем скомпрометировал его (или, например, его партнера, в зависимости от гарантий, предоставляемых протоколом). Также тривиальной можно считать «атаку», при которой противник не смог успешно завершить сеанс взаимодействия. Также протоколом могут предоставляться различные гарантии в случае компрометации *аутентифицирующей стороны*. Например, если протокол защищает от KCI-атак, то при компрометации абонента A не должно быть возможности представиться некоторым абонентом B перед A.

«Нетривиальной» можно считать такую атаку, в ходе которой противнику удалось незамеченно подменить какое-либо из передаваемых полей сообщений, от которых существенным образом зависит ход протокола или вырабатываемые в ходе протокола значения, навязать отправленное адресату A сообщение адресату $\mathsf{B} \neq \mathsf{A}$, в том числе успешно осуществить атаку отражения (reflection attack, см. [17]) либо совершить какие-либо ещё злонамеренные действия, реализация которых приводит к каким-либо нежелательным с точки зрения безопасности протокола последствиям.

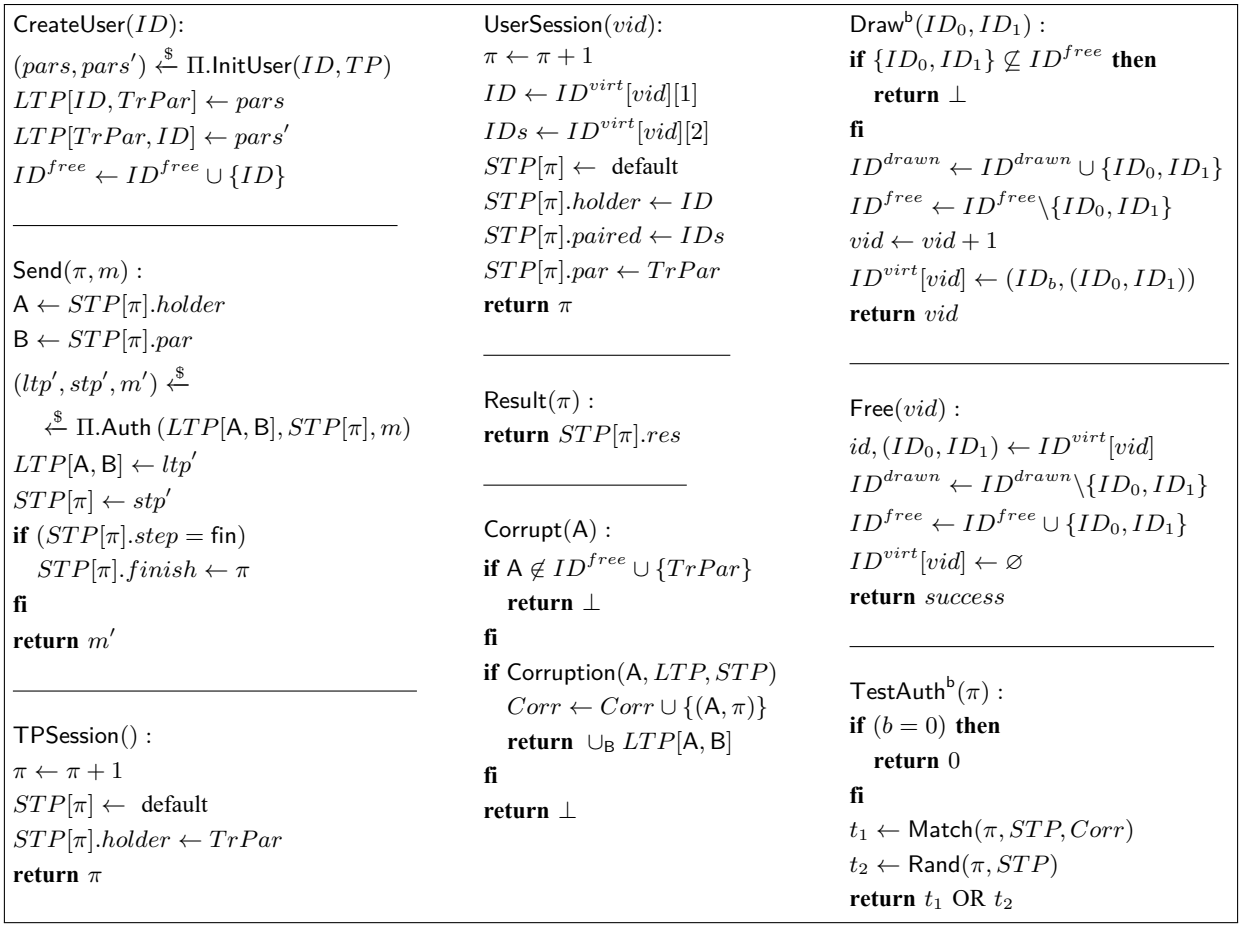

Рис. 2: Оракулы в эксперименте σAuth

Отметим также работу [31], которая посвящена сложности определения понятий «тривиальной» и «нетривиальной» атак для конкретного протокола.

Предикат Rand позволяет учесть атаки повтора: если противнику удастся добиться ситуации, в которой существуют два сеанса с одинаковым держателем и одинаковым партнером, в которых держатель сгенерировал одни и те же эфемерные значения, то это означает, что возможен повтор сообщений (атака повтора, replay-атака [16, 17]): противник может взять сообщения из «старого» сеанса и переслать их в «новый», оставаясь незамеченным. Также нам необходимо исключить ситуации, в которых такая коллизия маловероятна, но эфемерные значения являются «легко предсказуемыми» (т.н. «отложенные» (pre-play) атаки [16, 17]).

Замечание 7. *Иногда свойство исключения коллизий эфемерных значений исключают на этапе рассмотрения предиката* Match *(см., например, свойство (5) в определении партнеров-оракулов в работе [2]). В настоящей работе мы выделяем атаки повтора и отложенные атаки в отдельный класс с помощью предиката* Rand. *Таким образом, предикат* Rand *покрывает класс атак, связанных с некорректной генерацией эфемерных значений в сеансах связи, а предикат* Match *— иные уязвимости протокола, не связанные с эфемерными значениями.*

Предикат Corruption задает условие, проверяемое перед возможной компрометацией долговременного состояния участника. Так, протокол может подразумевать смену долговременного симметричного ключа через некоторое количество попыток аутентификации. Если некоторый участник уже пытался пройти аутентификацию на текущем ключе, то протокол может не предоставлять гарантий анонимности вплоть до смены ключа на производный. В таком случае предикат Corruption может проверять, участвовал ли атакуемый участник ID в некоторых сеансах на текущем ключе, и если участвовал, то запрещать компрометацию.

F. Технические ограничения в модели

Отметим некоторые технические ограничения, неявно предполагаемые в модели.

1) Оракул CreateUser принимает на вход уникальные значения ID: если ID ранее подавался на вход оракулу, т.е.
$$ID \in ID^{free} \cup ID^{drawn},$$
то возвращается символ ошибки $\perp$.
2) Оракулы Send, Result, TestAuth^b принимают на вход только ранее открытые сеансы π, в противном случае выдается ошибка.
3) Подаваемые на вход оракулу UserSession значения корректны:
$$ID^{virt}[vid] \neq \perp.$$

Указанные технические замечания не накладывают ограничения на множество рассматриваемых противников, поскольку «бесполезные» запросы всегда могут быть

исключены из рассмотрения без снижения вероятности успеха атаки в рамках рассматриваемой модели.

III. Обсуждение модели безопасности

Модель формализует возможности противника по взаимодействию с системой (задается с помощью оракулов), а также строго задает, что именно считается успешной атакой для рассматриваемого протокола (задается мерой успешности противника **Adv**).

A. О формализации возможностей противника

Опишем подробнее, как модель формализует возможности противника, перечисленные в разделе II-A. В рассматриваемой модели противник может выполнять следующие действия.

1) Создавать легитимных пользователей (оракул $\mathsf{CreateUser}(ID)$).
2) Запускать сеанс протокола аутентификации со стороны участника ДС (оракул TPSession).
3) Запускать сеанс протокола аутентификации со стороны некоторого *анонимизированного* абонента vid (оракул UserSession).
4) Пересылать сообщения участнику протокола (оракул Send). Сообщения пересылаются в рамках некоторого фиксированного сеанса π держателю сеанса.
5) Получать виртуальный идентификатор абонента vid (оракул Draw^b). Оракул $Draw^b$ в зависимости от бита b выбирает один из двух поданных на вход идентификаторов и ассоциирует с ним виртуальный идентификатор vid. По выданному виртуальному идентификатору можно открыть сеанс с *анонимизированным* абонентом. Для избежания возможности тривиальных атак оба абонента оказываются в связанном состоянии (их больше нельзя подавать на вход Draw вплоть до «освобождения»).
6) Для освобождения абонентов используется оракул Free. Оракул переводит пару «связанных» абонентов в «свободное» состояние, после чего каждого из них вновь можно анонимизировать в паре с некоторым абонентом ID.
7) Проверять результат сеанса π — в процессе, успех или ошибка (оракул $\mathsf{Result}(\pi)$).
8) Компрометировать участников, получая их текущее долговременное состояние (оракул $\mathsf{Corrupt}(ID)$).
9) Тестировать свойство аутентификации в сеансе π (оракул $\mathsf{TestAuth}^b$). В случае $b = 0$ оракул возвращает 0. В случае $b = 1$ оракул возвращает 1 в случае срабатывания одного из предикатов Match или Rand.

Замечание 8. *При рассмотрении* AKE*-протоколов к описанным выше оракулам обычно добавляется оракул раскрытия выработанного сеансового ключа* Reveal. *Оракул компрометации долговременного состояния можно дополнить оракулами компрометации и/или модификации промежуточных (эфемерных) значений (см. возможности противника в работе [20]).*

Также для некоторых протоколов важно различать онлайн- и оффлайн-атаки на протокол (например, PAKE*-протоколы [2]). В таком случае оракул активного вмешательства противника в общение* Send *может быть дополнен оракулом пассивного прослушивания канала* Exec *[32].*

Противник полностью контролирует всю сеть (в частности, может по своему усмотрению перемешивать, задерживать, пытаться видоизменить сообщения), что соответствует возможностям противника в модели Долева-Яо [19, раздел 2.3].

В терминологии, приведённой в работе [20], мы рассматриваем противника следующего вида:

- возможности взаимодействия на канал — класс C3 (задержка, модификация, замена, удаление, генерация сообщений в канале);
- возможности по регистрации абонентов — класс UR4 (навязывание ID абонентов при регистрации);
- возможности по взаимодействию с абонентами — класс $\mathsf{UA1} \cup \mathsf{UA2} \cup \mathsf{UA4} \cup \mathsf{UA5}$ (параллельные сеансы, запуск сеансов для выбранного пользователя и доверенной третьей стороны, компрометация долговременных ключей).

В модели σAuth при регистрации абонентов рассматривается только возможность навязывания ID — предполагается, что долговременные параметры централизованным и безопасным образом распределяются среди пользователей (например, на этапе производства).

B. О формализации требований

Опишем подробнее, как модель формализует требования, перечисленные в разделе II.

1) **Аутентификация участников:** противник не может успешно завершить протокол анонимной аутентификации без содействия легитимного участника. В противном случае (при соответствующем задании предиката Match) противник может создать успешно завершенный сеанс π без сопряженного к нему и протестировать сеанс π с помощью оракула TestAuth, тем самым достоверно определив бит b.
2) **Анонимность пользователей:** противник не получает никакой информации о том, какие именно абоненты взаимодействуют с системой (поведение абонентов и стенограммы сеансов различных абонентов неразличимы). В противном случае противник мог бы определить бит b по ответам оракула Draw (и в ходе взаимодействия с другими оракулами).

Заметим также, что в состав идентификатора абонента ID может включаться в том числе и аутентификационная информация. В таком случае неразличимость абонентов по их виртуальным идентификаторам свидетельствует в пользу того, что третья сторона не может получить никакой информации об аутентификационных данных аутентифицирующейся стороны.

В терминологии работы [30], в рассматриваемой модели путем различного задания набора предикатов

$$(\mathsf{Match}, \mathsf{Rand}, \mathsf{Corruption})$$

можно формализовать следующие свойства:

- C1, C7 — аутентификация участника протокола другим участником (предикат Match);
- C2 — аутентификация сообщений протокола (предикат Match);
- C3 — целостность сообщений протокола (предикат Match);
- C4 — защита от повторов (предикат Rand);

- C6 — групповая аутентификация (следует из C1);
- C8, C11, C12 — конфиденциальность ключа аутентификации, в том числе при компрометации (предикат Corruption);
- C15 — защита от навязывания параметров безопасности отправителя (предикат Match);
- C17 — инвариантность отправителя (предикат Match);
- C18 — анонимность абонентов (в несколько более общем смысле, чем указано в работе [30]).

C. Сравнение с другими моделями

Подход к формализации свойства аутентификации на основе «несуществования» сопряженного сеанса впервые предложен в работе [8] (отметим также альтернативный подход, предложенный в работе [2], в которой свойство аутентификации формализуется в виде невозможности противника отличить истинную аутентификационную метку от случайной строки аналогичной длины).

Подход к заданию свойства анонимности на основе неразличимости виртуальных идентификаторов предлагается (в контексте AKE-протоколов) в статьях [13, 14], однако, существуют и альтернативные модели, формализующие свойство анонимности на основе понятия *симулируемости* (см. работы [10, 11] и [12, 33]). В модели σAuth был выбран подход на основе виртуальных идентификаторов в связи с тем, что в таком виде рассматриваемое свойство проще совмещается с другими требуемыми свойствами, сформулированными в терминах подхода с Экспериментаторами.

Взаимодействие противника с участниками протокола происходит в рамках сеансов. Так, для того, чтобы послать сообщение некоторому пользователю, противник должен либо воспользоваться существующим открытым сеансом связи, либо открыть новый сеанс. Эта особенность исключает синтаксические атаки подобного вида: противник начинает сеанс с некоторым *анонимизированным* пользователем, а затем, в некоторый момент протокола посылает сообщение конкретному пользователю и смотрит, продолжит ли он общение.

D. Множество рассматриваемых атак

Рассмотрим типы атак, которые включены в модель σAuth и обычно упоминаются в контексте интерактивных протоколов (см., например, [4, 16, 17]).

- **Пассивные атаки.** Модель включает в себя противников, которые атакуют протокол лишь с использованием пассивного прослушивания канала связи (т.н. пассивные противники). Для каждого пассивного противника $\mathcal{A}$ можно построить активного противника $\mathcal{B}$ в исходной модели, который будет пересылать сообщения из одного сеанса в другой без их модификации и предоставлять стенограммы запуска протокола пассивному противнику $\mathcal{A}$. Все далее рассматриваемые типы атак являются активными.
- **Атаки повтора (replay-атаки):** модель включает в себя противников, которые атакуют протокол, пытаясь навязать ранее пересылаемые сообщения; для защиты от подобных атак в протокол могут «подмешиваться» случайности с обеих сторон взаимодействия, что обеспечивают уникальность стенограммы сеанса протокола. Заметим также, что в рамках модели исключение коллизий может отслеживаться с помощью предиката Rand.
- **Атаки с использованием параллельных сеансов:** модель включает в себя противников, которые атакуют протокол с использованием параллельных сеансов (по определению все участники протокола могут участвовать в параллельных сеансах связи); защита может обеспечиваться, например, путем «привязки» случайностей, сгенерированных обеими сторонами взаимодействия, к конкретному сеансу протокола путем вычисления значений некоторых псевдослучайных функций или функций выработки имитовставки от стенограммы переданных и полученных сообщений.
- **Атаки отражения (reflection):** модель включает в себя противников, которые атакуют протокол путем отражения сообщений (отправки сообщения участнику, который сам же и сформировал указанное сообщение); защита от подобных атак может достигаться путем внесения несимметричности в способ формирования полей сообщений.
- **Атаки подмены (параметров) протокола.**

Также следует упомянуть **атаки на отслеживание изменения внутреннего состояния,** актуальные в контексте анонимных протоколов. Допустим, что противник имеет возможность отслеживать изменение внутреннего состояния пользователя (без взлома внутреннего состояния). В таком случае он может использовать атаку следующего вида.

1) Противник взаимодействует с выбранным пользователем ID_0 путем подачи на вход оракулу Draw^b запроса (ID_0, ID_0) — в этом случае он уверен, что vid соответствует ID_0, и накапливает некоторую необходимую ему информацию. В конце взаимодействия противник «освобождает» пользователя $\mathsf{Free}(vid)$.
2) Противник выбирает второго пользователя $ID_1 \neq ID_0$, подает на вход оракулу Draw^b запрос (ID_0, ID_1) и взаимодействует с выбранным анонимизированным пользователем необходимое число раз. В конце взаимодействия противник «освобождает» пользователей — делает запрос к оракулу $\mathsf{Free}(vid)$.
3) После окончания взаимодействия противник вновь подает на вход оракулу Draw^b запрос (ID_0, ID_0). По ответам пользователя он выявляет, изменилось ли его внутреннее состояние с момента последнего сеанса или нет. Если оно изменилось предсказанным образом, то противник знает, что все это время он взаимодействовал с пользователем ID_0, а значит $b = 0$, в противном случае $b = 1$.

E. Возможные расширения модели

Можно указать несколько возможностей по дальнейшему расширению модели на более общие случаи.

1) В модели рассматривается случай (односторонней/взаимной) аутентификации некоторого абонента и фиксированной ДС (выделенный участник). Во многих протоколах аутентификации, используемых в настоящий момент, происходит аутентификация «равноправных» пользователей A и B друг перед

другом. При этом в протоколе также может присутствовать ДС для задачи распределения симметричных ключей перед началом взаимодействия или для «помощи» сторонам в ходе протокола (см., например, протокол Kerberos [19]). Модель может быть модифицирована для рассмотрения подобных случаев (см. замечание 4).

2) В модели рассматривается случай протоколов аутентификации с доверенным этапом регистрации. На практике процесс регистрации абонентов также может быть подвержен атакам. В таком случае необходимо также включать в модель процесс регистрации ключевых пар и рассматривать угрозы, специфические для процесса регистрации (класс возможностей противника AR в терминологии работы [20] и некоторые дополнительные возможности из класса UR, такие как навязывание несогласованной связанной ключевой пары при регистрации [34]; см. также замечание 8).
3) В модели не рассматриваются возможности компрометации/навязывания/повтора эфемерных значений, используемых в одном сеансе протокола (возможности UA6-UA15 в терминологии работы [20]).
4) В модели не рассматриваются атаки, при которых компрометируется абонент, участвующий в настоящий момент в некотором сеансе связи (перед компрометацией абонент должен быть «освобожден»).
5) В модели не рассматриваются дополнительные свойства, которые может обеспечивать протокол: так, например, в работе [9] рассматривается протокол аутентификации с передачей дополнительных данных, к которому предъявляются требования по обеспечению конфиденциальности и целостности передаваемых данных, а также по привязке процесса аутентификации к конкретным данным. Также из рассмотрения исключены некоторые специфические угрозы, например, атаки пересылки (relay-атаки, см. [27, 28, 29, 35]).

IV. Применение модели для оценки безопасности протокола аутентификации

Рассмотрим практическое применение модели σAuth для изучения свойств безопасности протокола аутентификации в рамках механизмов безопасной передачи идентификатора абонента (схема ECIES, см. [36, 37, 38]) и механизма выработки ключа 5G-AKA [14, 39, 40]. Мы опишем протокол Π, построенный на основе подпротокола аутентификации в рамках процедуры аутентификации и выработки общего ключа в сетях 5G. Затем покажем, как рассматриваемый протокол формализуется в рамках рассмотренной выше модели. Наконец, мы продемонстрируем атаки на протокол Π в рамках модели σAuth.

A. Используемые криптографические механизмы

Далее мы будем использовать несколько «базовых» криптографических механизмов.

Определение 5. *Псевдослучайной функцией будем называть семейство функций*

$$\mathsf{F} = \{\mathsf{F}_K \colon Dom \to \{0,1\}^n \mid K \in \{0,1\}^{klen}\},$$

индексированных ключом K из множества $\{0,1\}^{klen}$.

Определение 6. *Схемой аутентифицированного шифрования* AE *будем называть тройку (вероятностных) алгоритмов: (а) алгоритм генерации секретного ключа схемы шифрования* KGen*; (б) алгоритм зашифрования* $\mathsf{Enc}(K, m)$*; (в) алгоритм расшифрования* $\mathsf{Dec}(K, ct)$*. Алгоритм расшифрования возвращает либо некоторый открытый текст m, либо символ ошибки $\perp$. Схема* AE *должна удовлетворять требованию корректности: для любых m и $K \xleftarrow{\$}$ AE.KGen выполняется*

$$\mathsf{AE.Dec}(K, \mathsf{AE.Enc}(K, m)) = m.$$

Примером схемы аутентифицированного шифрования может служить алгоритм **MGM** [41] или режим шифрования **CTR** [42] совместно с алгоритмом выработки имитовставки **CMAC** [42] на независимых ключах в композиции вида **Encrypt-then-MAC** [43].

Определение 7. *Схемой выработки общего ключа* KE *будем называть пару алгоритмов: (а) алгоритм* PairGen*, возвращающий случайно выбранную ключевую пару (sk, pk); (б) алгоритм* Combine*, принимающий на вход два ключа — открытый pk и закрытый sk — и возвращающий выработанный ключ K. При этом должно выполняться стандартное требование корректности генерации ключей:*

$$(sk, pk) \xleftarrow{\$} \mathsf{KE.PairGen},\ \ (esk, epk) \xleftarrow{\$} \mathsf{KE.PairGen} \Rightarrow$$
$$\Rightarrow \mathsf{KE.Combine}(sk, epk) = \mathsf{KE.Combine}(esk, pk).$$

Примером схемы KE может служить схема **VKO** [44], анализ которой приведен в работе [18].

B. Описание изучаемого протокола Π

Пусть задан набор псевдослучайных функций $\mathsf{F}_1, \ldots, \mathsf{F}_5$, схема аутентифицированного шифрования AE и схема выработки общего ключа KE. В описании протокола Π мы исходим из следующих положений.

1) Абонент (**User**) обладает уникальным идентификатором ID.
2) Перед началом взаимодействия между абонентом и ДС распределен общий секретный ключ K_{ID}, а также открытый ключ ДС pk_{TP}, единый для всех абонентов.

Схема протокола Π приведена на рис. 3. Результатом работы протокола Π может являться одно из трех значений: (1) успешное завершение аутентификации accept; (2) ошибка аутентификации $\perp_{auth}$; (3) ошибка рассинхронизации $\perp_{sync}$.

В случае неудачи во время проверок ① или ④ (см. рис. 3) ДС прерывает выполнение протокола, результат работы протокола равен $\perp_{auth}$. В случае неудачи во время проверки ② абонент прерывает выполнение протокола, результат работы протокола равен $\perp_{auth}$. В случае неудачи во время проверки ③ абонент выполняет процедуру ресинхронизации (см. псевдокод на рис. 4), результат работы протокола равен $\perp_{sync}$.

Для упрощения анализа в подпроцедуру аутентификации в рамках исходного протокола 5G-AKA были внесены следующие изменения: был убран этап выработки общего ключа, а также изменено описание псевдослучайных функций, используемых в протоколе (например, многоступенчатая процедура выработки значения RES или способ нумерации функций).

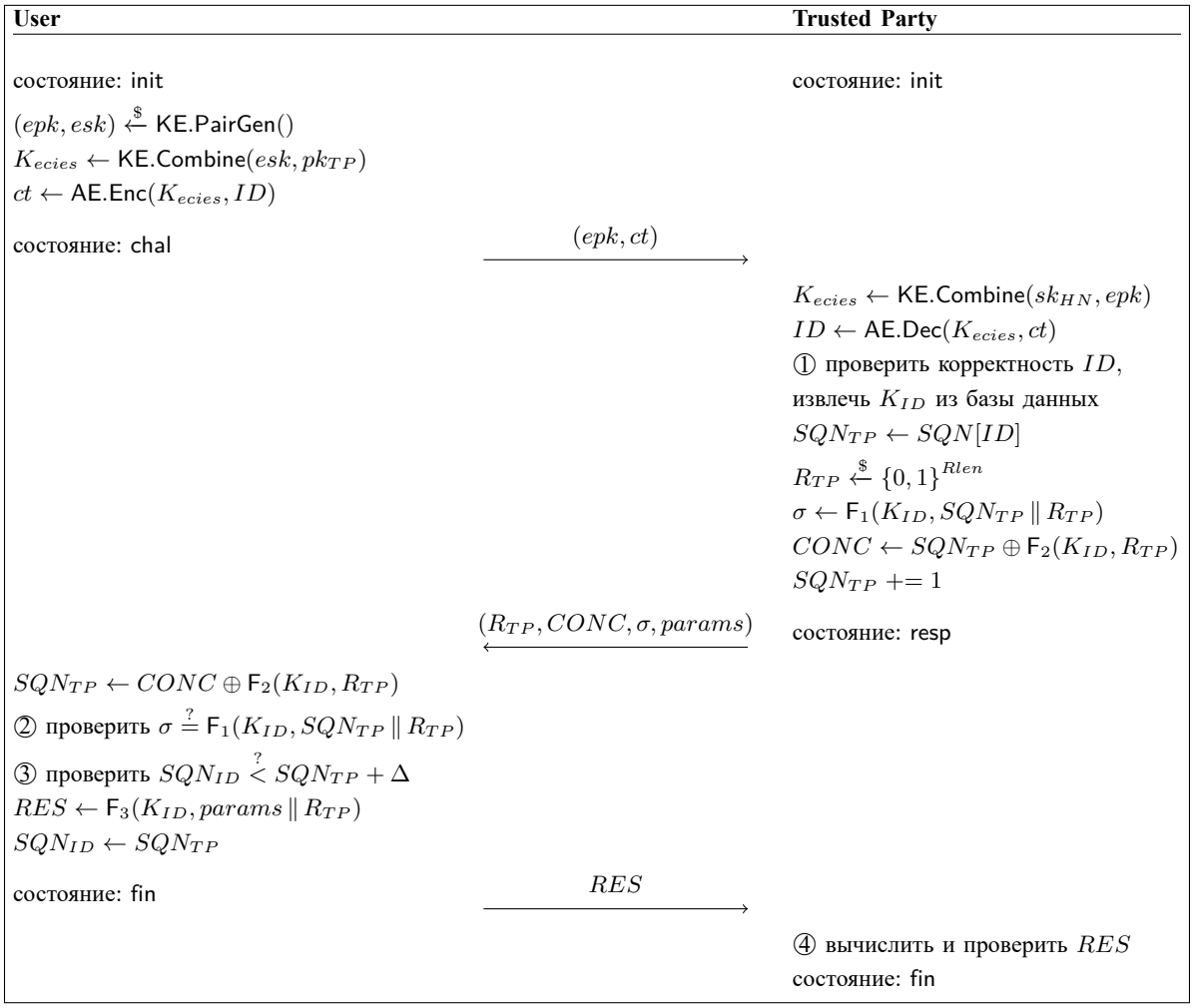

Рис. 3: Схема работы протокола Π

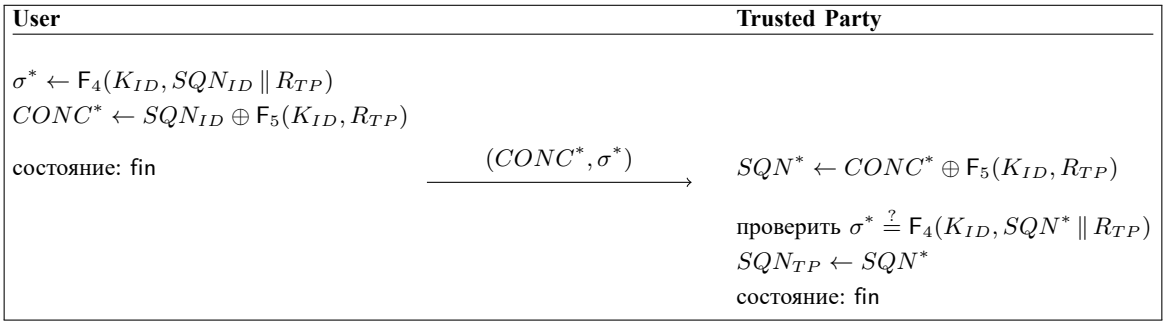

Рис. 4: Процедура ресинхронизации

C. Описание формализации для протокола Π

Зададим параметры модели σAuth для изучения безопасности протокола Π. Алгоритмы InitTP и InitUser для рассматриваемого протокола Π формализуют процедуру предварительного распределения ключей, во время которой стороны взаимодействия получают общий набор ключей (K_{ID}, pk_{TP}) и инициализируют счетчики $SQN_{ID} \leftarrow 0$, $SQN_{TP}[ID] \leftarrow 0$. Мы будем предполагать, что открытый ключ ДС pk_{TP} записывается во внутреннее состояние абонентов при выполнении алгоритма инициализации Π.InitUser(ID, TP).

Для рассматриваемого протокола Π долговременными параметрами участника A для связи с участником B $LTP[\mathsf{A}, \mathsf{B}]$ являются:

- идентификатор участника A,
- счетчик SQN,
- закрытый ключ ДС sk_{TP},
- открытый ключ ДС pk_{TP},
- секретный ключ K_{A}.

Алгоритм Π.InitTP генерирует долговременную ключевую пару ДС $TP \leftarrow (sk, pk)$ и возвращает её в качестве результата работы.

Алгоритм Π.InitUser(ID, TP) заполняет поля структуры долговременных параметров абонента ltp следующим образом:

$$ltp.id \leftarrow ID,\ \ ltp.SQN \leftarrow 0,\ \ ltp.K \xleftarrow{\mathcal{U}} \{0,1\}^{256},$$
$$ltp.sk \leftarrow \perp,\ \ ltp.pk \leftarrow TP.pk.$$

Для структуры ltp' отличается только поле $ltp'.sk$:

$$ltp' \leftarrow ltp,\ \ ltp'.sk \leftarrow TP.sk.$$

Для протокола Π стенограмма сеанса $STP[\pi]$ содержит, среди прочих, следующие поля: (а) эфемерный ключ epk; (б) шифртекст ct; (в) случайность R_{TP} со стороны ДС, (г) скрытое значение счетчика $CONC$, (д) параметры соединения $params$. Отметим, что все указанные поля передаются в открытом виде по каналам связи.

Предикат $\mathsf{Match}(\pi, STP, Corr)$ задается условием:

$$\mathsf{Match}(\pi, STP, Corr) \leftarrow (t_1 \text{ AND NOT } t_2)\,,$$

где условие t_1 определено следующим образом: результатом сеанса π является успешная аутентификация или ошибка синхронизации, держатель абонента и ДС не скомпрометированы:

$$t_1 = \mathsf{True} \Leftrightarrow \begin{cases} STP[\pi].res \in \{\mathsf{accept}, \perp_{sync}\}, \\ STP[\pi].holder \notin Corr, \\ TrPar \notin Corr, \end{cases}$$

а значение $t_2 = \mathsf{True}$ тогда и только тогда, когда выполнено одно из условий:

1) Если держатель сеанса π — абонент, то существует сеанс π' с держателем ДС, такой что

$$STP[\pi'].par = STP[\pi].holder,$$

и в стенограммах $STP[\pi].trans$ и $STP[\pi'].trans$ совпадают поля epk, CT, R_{TP}, $CONC$.
2) Если держатель сеанса π — ДС, и $STP[\pi].res = \mathsf{accept}$, то существует сеанс π' с держателем — абонентом, такой что

$$STP[\pi'].holder = STP[\pi].par,$$

и при этом выполнены условия:
- в стенограммах $STP[\pi].trans$ и $STP[\pi'].trans$ совпадают поля epk, CT, R_{TP}, $CONC$, $params$;
- $STP[\pi'].res = \mathsf{accept}$.

Замечание 9. *Если найдётся сеанс π', для которого выполняются условия, перечисленные при рассмотрении условия t_2, то такой сеанс называется сопряженным к сеансу π.*

Замечание 10. *Условие t_1 проверяет, что сеанс π «успешно завершен», и абонент-держатель или ДС не были скомпрометированы. Условие NOT t_2 проверяет, что не реализовалась атака в сеансе с абонентом, в ходе которой противник просто пересылает сообщения из одного сеанса в другой, не меняя никаким образом «существенные» поля. При этом условия на t_2 «несимметричны» относительно держателей сеанса, поскольку целостность переданного поля $params$ может быть проконтролирована только на стороне ДС.*

Предикат $\mathsf{Rand}(\pi, STP)$ истинен тогда и только тогда, когда выполнено одно из следующих условий.

1) Держатель сеанса π — абонент ID, и найдется сеанс $\pi' \neq \pi$, что для π и π' совпадают поля epk (в составе стенограммы) и $holder$ (коллизия честно сгенерированных значений epk).
2) Держатель сеанса π — абонент ID, и найдется сеанс $\pi' \neq \pi$ с держателем ДС, предполагаемым партнером ID, в сеансах π и π' совпадает поле epk (в составе стенограммы) и выполняется условие $Session[\pi'].finish < Session[\pi].start$ (предсказание значения epk до его генерации).
3) Держатель сеанса π — ДС, и найдется сеанс $\pi' \neq \pi$, что для сеансов π и π' совпадают поля R_{TP} (в составе стенограммы) и $holder$ (коллизия честно сгенерированных значений R_{TP}).
4) Держатель сеанса π — ДС, и найдется сеанс $\pi' \neq \pi$ с держателем абонентом, для сеансов π и π' совпадает поле R_{TP} (в составе стенограммы) и выполняется условие $Session[\pi'].finish < Session[\pi].start$ (предсказание значения R_{TP} до его генерации).

В рассматриваемом протоколе аутентификация производится на основе долговременного симметричного ключа K_{ID}, смена которого в ходе взаимодействия участников не подразумевается. Таким образом, при компрометации абонента мы не можем обеспечить никаких свойств — в частности, нарушается свойство анонимности в уже завершенных сеансах. Следовательно, для рассматриваемого протокола Π предикат Corruption не должен позволять взламывать абонентов, которые участвовали хотя бы в одном из сеансов. Также мы должны запретить компрометацию абонентов, которые сами в сеансе не участвовали, но были связаны с абонентом, который участвовал в каком-либо сеансе связи — в противном случае в рамках модели существует тривиальная атака на свойство анонимности: необходимо провести сеанс с анонимизированным абонентом $vid \leftrightarrow (ID_0, ID_1)$, а затем последовательно попытаться взломать ID_0 и ID_1. Таким образом,

$$\mathsf{Corruption}(\mathsf{A}, LTP, STP) = \mathsf{False} \Leftrightarrow$$
$$\Leftrightarrow (\exists\, \pi : \mathsf{A} \in STP[\pi].paired) \text{ OR } \mathsf{A} = TrPar.$$

V. Анализ угроз атак повтора в протоколе Π

Описанный протокол Π уязвим ко множеству атак. Приведем некоторые из них в рамках модели σAuth.

A. Атака проверки ID

Протокол Π не обеспечивает свойство анонимности абонентов. Противник может самостоятельно формировать значения (epk', ct') и подменять (epk, ct) в ходе работы протокола, тем самым проверяя, является ли атакуемый анонимизированный абонент некоторым конкретным абонентом ID'. Опишем атаку более формально.

1) Противник делает два запроса к оракулу CreateUser и создает двух пользователей $ID_0 \neq ID_1$.
2) Противник делает запрос (ID_0, ID_1) к оракулу Draw^b, получает vid.
3) Противник запускает сеанс π_1 со стороны анонимизированного абонента vid и получает от него пару (epk, ct), где

$$ct \xleftarrow{\$} \mathsf{AE.Enc}(K, ID_b), K \leftarrow \mathsf{KE.Combine}(esk, pk).$$

4) Противник генерирует ключевую пару $(esk', epk') \xleftarrow{\$} \mathsf{KE.PairGen}$ и зашифровывает ID_0 на ключе K':

$$ct' \xleftarrow{\$} \mathsf{AE.Enc}(K', ID_0),\ K' \leftarrow \mathsf{KE.Combine}(esk', pk).$$

5) Противник открывает сеанс π_2 со стороны ДС, посылает в рамках сеанса π_2 пару (epk', ct'), получает ответ $(R_{TP}, CONC, \sigma, params)$ и пересылает его в сеанс π_1.
6) Если анонимизированный абонент не прерывает сеанс и продолжает общение с противником, то противник выдает $b' = 0$ в качестве ответа в эксперименте; в противном случае (сеанс прерывается ошибкой) — выдает $b' = 1$.

Если $b = 0$, то противник корректно выдает ответ $b' = 0$. Если $b = 1$, то противник может некорректно выдать ответ $b' = 0$ в случае, если случайно выполнилось одно из двух условий (1), (2):

$$K_{ID_0} = K_{ID_1}, \tag{1}$$

$$K_{ID_0} \neq K_{ID_1}, \quad \mathsf{F}_1(K_{ID_0}, SQN_{TP}[ID_0] \,\|\, R_{TP}) = \mathsf{F}_1(K_{ID_1}, SQN_{TP}[ID_1] \,\|\, R_{TP}). \tag{2}$$

Каждое из указанных условий выполняется с пренебрежимо малой вероятностью. Атака возможна в силу того, что поля сеанса защищаются протоколом «неравномерно» (нет привязки первого сообщения к последующим сообщениям в протоколе). Перепишем атаку в виде псевдокода противника $\mathcal{A}$ (рис. 5).

$\mathcal{A}(pk)$
$\mathsf{CreateUser}(ID_0)$
$\mathsf{CreateUser}(ID_1)$
$vid \leftarrow \mathsf{Draw}^b(ID_0, ID_1)$
$\pi_1 \leftarrow \mathsf{UserSession}(vid)$
$(epk, ct) \leftarrow \mathsf{Send}(\pi_1, \perp)$
$(esk', epk') \xleftarrow{\$} \mathsf{KE.PairGen}$
$K' \leftarrow \mathsf{KE.Combine}(esk', pk)$
$ct' \xleftarrow{\$} \mathsf{AE.Enc}(K', ID_0)$
$\pi_2 \leftarrow \mathsf{TPSession}()$
$(R_{TP}, CONC, \sigma, params) \leftarrow \mathsf{Send}(\pi_2, (epk', ct'))$
$RES \leftarrow \mathsf{Send}(\pi_1, (R_{TP}, CONC, \sigma, params))$
$tmp \leftarrow \mathsf{Result}(\pi_1)$
if $tmp = \perp_{auth}$
  return 1
else
  return 0

Рис. 5: Псевдокод атаки проверки ID в рамках модели σAuth

B. Атака проверки (epk', ct')

Аналогичным образом можно поступить, если противник хочет проверить, один и тот же абонент участвует в текущем сеансе и в некотором прошлом, или же это разные абоненты.

На первом шаге атаки противник проводит сеанс связи с абонентом ID_0 с помощью запроса (ID_0, ID_0) к оракулу Draw^b, и в сеансе с полями (epk', ct') запоминает значения $(R'_{TP}, CONC', \sigma', params')$, где $CONC'$ — скрытое значение идентификатора SQN' для ID_0. На втором шаге атаки противник проводит Δ сеансов связи между ID_0 и ДС, тем самым наращивая счетчик SQN со стороны ID_0 и TP:

$$SQN \to SQN + \Delta.$$

На третьем шаге противник открывает сеанс связи с анонимизированным абонентом, подавая на вход оракулу Draw^b пару (ID_0, ID_1). Получив от анонимизированного абонента пару (epk, ct), противник посылает ему в ответ четверку $(R'_{TP}, CONC', \sigma', params')$.

Если $b = 0$, то vid (соответствует абоненту ID_0) расшифрует «старое» значение $SQN'_{TP}[ID_0]$, которое не проходит проверку $SQN_{ID} < SQN'_{TP} + \Delta$, и vid заканчивает сеанс с ошибкой ресинхронизации $\perp_{sync}$.

Если $b = 1$, то vid (соответствует абоненту ID_1) некорректно расшифровывает значение SQN' (на своем ключе K_{ID_1}), в результате чего получает некоторое значение $\widetilde{S}$, а затем проверяет значение имитовставок

$$\sigma = \mathsf{F}_1(K_{ID_0}, SQN' \,\|\, R'_{TP}) \stackrel{?}{=} \mathsf{F}_1(K_{ID_1}, \widetilde{S} \,\|\, R'_{TP}),$$

которое выполняется с пренебрежимо малой вероятностью, и vid заканчивает сеанс с ошибкой проверки имитовставки $\perp_{auth}$.

Таким образом, поведение абонентов в рассмотренных случаях различно, код ошибки может быть получен противником при обращении к оракулу Result (или выведен неявным образом по структуре ответа, в процедуре ресинхронизации абонентом посылаются дополнительные данные), что позволяет нарушать анонимность. Таким образом можно проверить, один и тот же абонент участвовал в разных сеансах связи или различные. Перепишем атаку в виде псевдокода противника $\mathcal{A}$ (рис. 6).

Атака возможна в силу того, что поля сеанса защищаются протоколом «неравномерно» (нет привязки первого сообщения к последующим сообщениям в протоколе), в частности, возможен повтор «старых» сообщений, которые будут корректно (в соответствии с протоколом) обработаны второй стороной. Заметим также, что атака является частным случаем атаки на отслеживание изменения внутреннего состояния (см. раздел V).

VI. Заключение

В настоящей работе представлена модель безопасности σAuth, формализующая свойство анонимной аутентификации для интерактивных протоколов аутентификации с выделенной доверенной стороной и безопасной процедурой инициализации абонентов. Рассмотрены различные свойства модели: как модель задает возможности противника и формализует изначальные требования к подобным протоколам, какое множество атак покрывается моделью. Приведено сравнение с другими моделями для (P)AKE-протоколов и протоколов аутентификации. Для протокола Π, построенного на основе механизма 5G-AKA, показаны атаки в предложенной модели, нарушающие свойство анонимности.

Авторы благодарят Л. Р. Ахметзянову за полезные обсуждения и внимательное отношение к работе, благодаря которым текст работы был значительно улучшен.

$\mathcal{A}(pk)$

$\mathsf{CreateUser}(ID_0)$
$vid \leftarrow \mathsf{Draw}^b(ID_0, ID_0)$
$\pi_1 \leftarrow \mathsf{UserSession}(vid)$
$(epk', ct') \leftarrow \mathsf{Send}(\pi_1, \perp)$
$\pi_2 \leftarrow \mathsf{TPSession}()$
$(R'_{TP}, CONC', \sigma', params') \leftarrow \mathsf{Send}(\pi_2, (epk', ct'))$
repeat Δ **times**
  $\pi_1 \leftarrow \mathsf{UserSession}(vid)$
  $\pi_2 \leftarrow \mathsf{TPSession}()$
  $(epk, ct) \leftarrow \mathsf{Send}(\pi_1, \perp)$
  $(R_{TP}, CONC, \sigma, params) \leftarrow \mathsf{Send}(\pi_2, (epk, ct))$
  $RES \leftarrow \mathsf{Send}(\pi_1, (R_{TP}, CONC, \sigma, params))$
  $\mathsf{Send}(\pi_2, RES)$
$\mathsf{Free}(vid)$
$\mathsf{CreateUser}(ID_1)$
$vid \leftarrow \mathsf{Draw}^b(ID_0, ID_1)$
$\pi_1 \leftarrow \mathsf{UserSession}(vid)$
$(epk, ct) \leftarrow \mathsf{Send}(\pi_1, \perp)$
$RES' \leftarrow \mathsf{Send}(\pi_1, (R'_{TP}, CONC', \sigma', params'))$
$tmp \leftarrow \mathsf{Result}(\pi_1)$
if $tmp = \perp_{sync}$
  return 0
else // $tmp = \perp_{auth}$
  return 1

Рис. 6: Псевдокод атаки проверки (epk', ct') в рамках модели σAuth

Библиография

[1] A cryptographic analysis of the TLS 1.3 handshake protocol / B. Dowling, M. Fischlin, F. Gunther, D. Stebila // Journal of Cryptology. — 2021. — Vol. 34, no. 4. — P. 1–69.

[2] Алексеев Е. К., Смышляев С. В. О безопасности протокола SESPAKE // Прикладная дискретная математика. — 2020. — no. 50. — P. 5–41.

[3] Boyd C., Mathuria A., Stebila D. Protocols for authentication and key establishment. — Springer, 2003.

[4] Choo K. K. R. Secure key establishment. — Springer Science & Business Media, 2008.

[5] Brzuska C. On the foundations of key exchange. — 2012.

[6] Fischlin M., Gunther F. Multi-stage key exchange and the case of Google's QUIC protocol // Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. — 2014. — P. 1193–1204.

[7] Key Confirmation in Key Exchange: A Formal Treatment and Implications for TLS 1.3 / M. Fischlin, F. Günther, B. Schmidt, B. Warinschi // 2016 IEEE Symposium on Security and Privacy (SP). — 2016. — P. 452–469.

[8] Bellare M., Rogaway P. Entity authentication and key distribution // Annual international cryptology conference / Springer. — 1993. — P. 232–249.

[9] On the security of one RFID authentication protocol / A. Chichaeva, S. Davydov, E. Griboedova, K. Tsaregorodtsev // The 12th Workshop on Current Trends in Cryptology (CTCrypt 2023). — 2023.

[10] Vaudenay S. On privacy models for RFID // International conference on the theory and application of cryptology and information security / Springer. — 2007. — P. 68–87.

[11] Paise R. I., Vaudenay S. Mutual authentication in RFID: security and privacy // Proceedings of the 2008 ACM symposium on Information, computer and communications security. — 2008. — P. 292–299.

[12] A zero-knowledge based framework for RFID privacy / R. H. Deng, Y. Li, M. Yung, Y. Zhao // Journal of Computer Security. — 2011. — Vol. 19, no. 6. — P. 1109–1146.

[13] A new RFID privacy model / J. Hermans, A. Pashalidis, F. Vercauteren, B. Preneel // European symposium on research in computer security / Springer. — 2011. — P. 568–587.

[14] Koutsos A. The 5G-AKA Authentication Protocol Privacy. — 2019. — 06. — P. 464–479.

[15] The privacy of the TLS 1.3 protocol / G. Arfaoui, X. Bultel, P. A. Fouque et al. // Proceedings on Privacy Enhancing Technologies. — 2019. — Vol. 2019. — P. 190–210.

[16] Основы криптографии / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. — М. : Гелиос АРВ, 2005. — P. 480. — 3-е изд., испр. и доп.

[17] Черемушкин А. В. Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. Приложение. — 2009. — no. 2. — P. 115–150.

[18] О криптографических свойствах алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 / Е. К. Алексеев, И. Б. Ошкин, В. О. Попов, С. В. Смышляев // Математические вопросы криптографии. — 2016. — Vol. 7, no. 1. — P. 5–38.

[19] Мао В. Современная криптография: теория и практика. — Издательский дом Вильямс, 2005.

[20] Alekseev E., Kyazhin S. Probing the security landscape for authenticated key establishment protocols // The 12th Workshop on Current Trends in Cryptology (CTCrypt 2023). — 2023.

[21] Katz J., Lindell Y. Introduction to modern cryptography. — CRC press, 2020.

[22] Mittelbach A., Fischlin M. The theory of hash functions and random oracles. An Approach to Modern Cryptography. — Springer Cham, 2021.

[23] On symmetric encryption with distinguishable decryption failures / A. Boldyreva, J. P. Degabriele, K. G. Paterson, M. Stam // Fast Software Encryption: 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers 20 / Springer. — 2014. — P. 367–390.

[24] New privacy issues in mobile telephony: fix and verification / M. Arapinis, L. Mancini, E. Ritter et al. // Proceedings of the 2012 ACM conference on Computer and communications security. — 2012. — P. 205–216.

[25] New privacy threat on 3G, 4G, and upcoming 5G-AKA protocols / R. Borgaonkar, L. Hirschi, S. Park, A. Shaik // Cryptology ePrint Archive. — 2018.

[26] Alekseev E., Kyazhin S., Smyshlyaev S. The threat of forcing the identical roles for authenticated key establishment protocols // Journal of Computer Virology and Hacking Techniques. — 2024. — Vol. 20, no. 2. — P. 225–230.

[27] Brands S., Chaum D. Distance-bounding protocols // Workshop on the Theory and Application of of Cryptographic Techniques / Springer. — 1993. — P. 344–359.

[28] So near and yet so far: Distance-bounding attacks in wireless networks / J. Clulow, G. P. Hancke, M. G. Kuhn, T. Moore // European Workshop on Security in Ad-hoc and Sensor Networks / Springer. — 2006. — P. 83–97.

[29] Security of distance-bounding: A survey / G. Avoine, M. A. Bingöl, I. Boureanu et al. // ACM Computing Surveys (CSUR). — 2018. — Vol. 51, no. 5. — P. 1–33.

[30] Нестеренко А. Ю., Семенов А. М. Методика оценки безопасности криптографических протоколов // Прикладная дискретная математика. — 2022. — no. 56. — P. 33–82.

[31] Li Y., Schäge S. No-match attacks and robust partnering definitions: defining trivial attacks for security protocols is not trivial // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. — 2017. — P. 1343–1360.

[32] Bellare M., Pointcheval D., Rogaway P. Authenticated key exchange secure against dictionary attacks // International conference on the theory and applications of cryptographic techniques / Springer. — 2000. — P. 139–155.

[33] A new framework for RFID privacy / R. H. Deng, Y. Li, M. Yung, Y. Zhao // European Symposium on Research in Computer Security / Springer. — 2010. — P. 1–18.

[34] Алексеев Е. К., Кяжин С. Н., Смышляев С. В. Атаки на протоколы аутентифицированной выработки общего ключа при навязывании будущих открытых эфемерных ключей // Прикладная дискретная математика. — 2024. — Vol. 66. — P. 60–77.

[35] Beth T., Desmedt Y. Identification tokens—or: Solving the chess grandmaster problem // Conference on the Theory and Application of Cryptography / Springer. — 1990. — P. 169–176.

[36] Abdalla M., Bellare M., Rogaway P. The oracle Diffie-Hellman assumptions and an analysis of DHIES // Topics in Cryptology—CT-RSA 2001: The Cryptographers' Track at RSA Conference 2001 San Francisco, CA, USA, April 8–12, 2001 Proceedings / Springer. — 2001. — P. 143–158.

[37] Smart N. P. The exact security of ECIES in the generic group model // IMA International Conference on Cryptography and Coding / Springer. — 2001. — P. 73–84.

[38] Царегородцев К. Д. Свойства конфиденциальности и целостности схемы ECIES // Математические вопросы криптографии. — 2024. — Vol. 15, no. 2. — P. 101–136.
[39] 3GPP TS 33.501 V 18.0.0. Security architecture and procedures for 5G system.
[40] Бельский В. С., Дрынкин А. В., Давыдов С. А. Вопросы обеспечения безопасности абонентов в сетях радиодоступа пятого поколения // International Journal of Open Information Technologies. — 2021. — Vol. 9, no. 7. — P. 32–54.
[41] Рекомендации по стандартизации Р 1323565.1.026-2019. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров, реализующие аутентифицированное шифрование. — 2019.
[42] Межгосударственный стандарт ГОСТ 34.13-2018 Информационная технология (ИТ). Криптографическая защита информации. Режимы работы блочных шифров. — 2018.
[43] Bellare B., Namprempre C. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm // Journal of Cryptology. — 2000. — Vol. 21. — P. 469–491.
[44] Рекомендации по стандартизации Р50.1.113-2016 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. — 2016.

On one method of formalizing the anonymous authentication property

A. Bakharev, V. Belsky, I. Gerasimov, K. Tsaregorodtsev

***Abstract*—In this paper, we propose a method for formalizing the anonymous authentication property based on the "provable security"paradigm. The article presents the pseudocode of the model, as well as comments regarding the potential capabilities of an attacker that are taken into account in the model and the security properties that it describes. A number of attacks that can be formalized using the model are considered, a comparison is made with similar models for (P)AKE protocols, and directions for further model extensions are highlighted. Finally, a step-by-step description of the formalization process is provided using the example of a specific authentication protocol, which is based on the authentication key agreement procedure in 5G networks.**

***Keywords*—identification, authentication, anonymity, privacy, provable security**

References

[1] A cryptographic analysis of the TLS 1.3 handshake protocol / B. Dowling, M. Fischlin, F. Gunther, D. Stebila // Journal of Cryptology. — 2021. — Vol. 34, no. 4. — P. 1–69.

[2] On the security of one password authenticated key exchange protocol / S. V. Smyshlyaev, I. B. Oshkin, E. K. Alekseev, L. R. Ahmetzyanova // Cryptology ePrint Archive. — 2015.

[3] Boyd C., Mathuria A., Stebila D. Protocols for authentication and key establishment. — Springer, 2003.

[4] Choo K. K. R. Secure key establishment. — Springer Science & Business Media, 2008.

[5] Brzuska C. On the foundations of key exchange. — 2012.

[6] Fischlin M., Gunther F. Multi-stage key exchange and the case of google's QUIC protocol // Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. — 2014. — P. 1193–1204.

[7] Key Confirmation in Key Exchange: A Formal Treatment and Implications for TLS 1.3 / M. Fischlin, F. Günther, B. Schmidt, B. Warinschi // 2016 IEEE Symposium on Security and Privacy (SP). — 2016. — P. 452–469.

[8] Bellare M., Rogaway P. Entity authentication and key distribution // Annual international cryptology conference / Springer. — 1993. — P. 232–249.

[9] On the security of one RFID authentication protocol / A. Chichaeva, S. Davydov, E. Griboedova, K. Tsaregorodtsev // The 12th Workshop on Current Trends in Cryptology (CTCrypt 2023). — 2023.

[10] Vaudenay S. On privacy models for RFID // International conference on the theory and application of cryptology and information security / Springer. — 2007. — P. 68–87.

[11] Paise R. I., Vaudenay S. Mutual authentication in RFID: security and privacy // Proceedings of the 2008 ACM symposium on Information, computer and communications security. — 2008. — P. 292–299.

[12] A zero-knowledge based framework for RFID privacy / R. H. Deng, Y. Li, M. Yung, Y. Zhao // Journal of Computer Security. — 2011. — Vol. 19, no. 6. — P. 1109–1146.

[13] A new RFID privacy model / J. Hermans, A. Pashalidis, F. Vercauteren, B. Preneel // European symposium on research in computer security / Springer. — 2011. — P. 568–587.

[14] Koutsos A. The 5G-AKA Authentication Protocol Privacy. — 2019. — 06. — P. 464–479.

[15] The privacy of the TLS 1.3 protocol / G. Arfaoui, X. Bultel, P. A. Fouque et al. // Proceedings on Privacy Enhancing Technologies. — 2019. — Vol. 2019. — P. 190–210.

[16] Osnovy kriptografii [Foundations of Cryptography] / A. P. Alferov, A. Yu. Zubov, A. S. Kuz'min, A. V. Cheremushkin. — Moscow : Helios, Association of Russian Universities, 2005. — P. 480. — In Russian.

[17] Cheremushkin A. V. Cryptographic protocols: main properties and vulnerabilities // Prikladnaya Diskretnaya Matematika. Supplement. — 2009. — no. 2. — P. 115–150.

[18] On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012 / E. K. Alekseev, I. B. Oshkin, V. O. Popov, S. V. Smyshlyaev // Mathematical Aspects of Cryptography. — 2016. — Vol. 7, no. 1. — P. 5–38.

[19] Mao W. Modern Cryptography: Theory and Practice. — Williams publishing, 2005. — In Russian.

[20] Alekseev E., Kyazhin S. Probing the security landscape for authenticated key establishment protocols // The 12th Workshop on Current Trends in Cryptology (CTCrypt 2023). — 2023.

[21] Katz J., Lindell Y. Introduction to modern cryptography. — CRC press, 2020.

[22] Mittelbach A., Fischlin M. The theory of hash functions and random oracles. An Approach to Modern Cryptography. — Springer Cham, 2021.

[23] On symmetric encryption with distinguishable decryption failures / A. Boldyreva, J. P. Degabriele, K. G. Paterson, M. Stam // Fast Software Encryption: 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers 20 / Springer. — 2014. — P. 367–390.

[24] New privacy issues in mobile telephony: fix and verification / M. Arapinis, L. Mancini, E. Ritter et al. // Proceedings of the 2012 ACM conference on Computer and communications security. — 2012. — P. 205–216.

[25] New privacy threat on 3G, 4G, and upcoming 5G AKA protocols / R. Borgaonkar, L. Hirschi, S. Park, A. Shaik // Cryptology ePrint Archive. — 2018.

[26] Alekseev E., Kyazhin S., Smyshlyaev S. The threat of forcing the identical roles for authenticated key establishment protocols // Journal of Computer Virology and Hacking Techniques. — 2024. — Vol. 20, no. 2. — P. 225–230.

[27] Brands S., Chaum D. Distance-bounding protocols // Workshop on the Theory and Application of of Cryptographic Techniques / Springer. — 1993. — P. 344–359.

[28] So near and yet so far: Distance-bounding attacks in wireless networks / J. Clulow, G. P. Hancke, M. G. Kuhn, T. Moore // European Workshop on Security in Ad-hoc and Sensor Networks / Springer. — 2006. — P. 83–97.

[29] Security of distance-bounding: A survey / G. Avoine, M. A. Bingöl, I. Boureanu et al. // ACM Computing Surveys (CSUR). — 2018. — Vol. 51, no. 5. — P. 1–33.

[30] Nesterenko A. Yu., Semenov A. M. Methodology for assessing the security of cryptographic protocols // Prikladnaya Diskretnaya Matematika. — 2022. — no. 56. — P. 33–82. — In Russian.

[31] Li Y., Schäge S. No-match attacks and robust partnering definitions: defining trivial attacks for security protocols is not trivial // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. — 2017. — P. 1343–1360.

[32] Bellare M., Pointcheval D., Rogaway P. Authenticated key exchange secure against dictionary attacks // International conference on the theory and applications of cryptographic techniques / Springer. — 2000. — P. 139–155.

[33] A new framework for RFID privacy / R. H. Deng, Y. Li, M. Yung, Y. Zhao // European Symposium on Research in Computer Security / Springer. — 2010. — P. 1–18.

[34] Alekseev E. K., Kyazhin S. N., Smyshlyaev S. V. Forcing future public ephemeral keys to attack authenticated key establishment protocols // Prikladnaya Diskretnaya Matematika. — 2024. — Vol. 66. — P. 60–77.

[35] Beth T., Desmedt Y. Identification tokens—or: Solving the chess grandmaster problem // Conference on the Theory and Application of Cryptography / Springer. — 1990. — P. 169–176.

[36] Abdalla M., Bellare M., Rogaway P. The oracle Diffie-Hellman assumptions and an analysis of DHIES // Topics in Cryptology— CT-RSA 2001: The Cryptographers' Track at RSA Conference 2001 San Francisco, CA, USA, April 8–12, 2001 Proceedings / Springer. — 2001. — P. 143–158.
[37] Smart N. P. The exact security of ECIES in the generic group model // IMA International Conference on Cryptography and Coding / Springer. — 2001. — P. 73–84.
[38] Tsaregorodtsev K. On the confidentiality and integrity of ECIES scheme // Mathematical Aspects of Cryptography. — 2024. — Vol. 15, no. 2. — P. 101–136.
[39] 3GPP TS 33.501 V 18.0.0. Security architecture and procedures for 5G system.
[40] Belsky V., Drynkin A., Davydov S. A subscriber's privacy on the 5G radio interface // International Journal of Open Information Technologies. — 2021. — Vol. 9, no. 7. — P. 32–54.
[41] R 1323565.1.026-2019. Information technology. Cryptographic data security. Block cipher modes implementing authenticated encryption. — 2019.
[42] GOST 34.13-2018. Information technology. Cryptographic data security. Modes of operation for block ciphers. — 2018.
[43] Bellare B., Namprempre C. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm // Journal of Cryptology. — 2000. — Vol. 21. — P. 469–491.
[44] R 50.1.113-2016. Information technology. Cryptographic data security. Cryptographic algorithms accompanying the application of electronic digital signature and hash-function. — 2016.