

Формальный анализ модифицированной версии протокола WireGuard на основе отечественных криптографических механизмов

Кирилл Царегородцев
Марина Скоробогатова

Компания «Актив»

Содержание раздела

- 1 Формальная верификация в целом
- 2 Протокол RU-Wireguard
- 3 Полученные результаты

Отличия от теоретико-сложностного подхода

- Инструменты формальной верификации оперируют на символьном [1] (теоретико-логическом) уровне.
- Основные составляющие:
 - **термы** — ключ, сообщение, случайное значение;
 - **функциональные символы** — Enc , Dec , H , ...;
 - **набор тождеств**, например: $\text{Dec}(K, \text{Enc}(K, M)) = M$.
- Все используемые криптографические механизмы являются идеальными черными ящиками (даже «более идеальными», чем модель случайного оракула или идеального блочного шифра).

[1] кроме таких инструментов, как CryptoVerif, EasyCrypt, см. далее

О различии инструментов

Подход	Теоретико-сложностной	Символьный (логический)
Базовые элементы	Двоичные строки, полином. вероятн. алгоритмы	Термы, функциональные символы
Противник	Полином. вероятн. алгоритм	Модель Долева-Яо
Результат	Вероятность успеха, преимущество	Бинарная величина 0/1
К чему стремимся	Неразличимость (indistinguishability), симулируемость (UC, simulability), неподделываемость (unforgeability)	Невозможность достигнуть «плохого» состояния, неотличимость процессов (observational equivalence)

О различии инструментов

Подход	Теоретико-сложностной	Символьный (логический)
Неполное знание	Сокращает перебор, повышает вероятность успеха	Не влияет
Допустимые преобразования	На основе неразличимости	На основе набора тождеств
Аппарат	Редукционизм (сведение)	Переписывание (rewriting), логический вывод
Когда анализ завершается	Стойкость сведена к «базовым» задачам	Применение правил не расширяет состояние (неподвижная точка)

Примеры различий

- Найти прообраз хэш-функции невозможно (принципиально невозможно, такое правило обычно отсутствует в системе тождеств).
- Коллизии «свежих» (случайно генерируемых) значений принципиально невозможны.
- Если противнику неизвестен хотя бы один бит ключа, то расшифрование невозможно.

Пример плохого протокола

Для аутентификации на основе симметричного ключа посылаем первые 255 бит ключа и меняем ключ на производный с обеих сторон взаимодействия.

Проблема останова и верификация

- Верификация криптографических свойств протоколов в общем случае — алгоритмически неразрешимая задача (см., например, различные примеры в [2]).
- На практике это означает, что верификация протокола может работать неограниченно долго, «пожирая» время и память — и мы не будем знать, остановится этот процесс или нет.
- Фактически разработчики инструментов идут по одному из нескольких возможных путей [3]: ограничение на число сеансов (bounded/unbounded); ограничение на фрагменты логики; упрощение протокола; ассистирование человека (human-assisted): эвристики, интерактивный режим.

[2] R. Chadha и др., «Automated verification of equivalence properties of cryptographic protocols».

[3] M. Barbosa и др., «SoK: Computer-aided cryptography».

Ограничения

- Мы можем ограничить общность рассматриваемых криптографических протоколов, сузив возможные механизмы до некоторого заранее определенного подмножества криптографических операций (фрагмента логики); теряем в **выразительности**.
- Вместо исходного протокола мы можем рассматривать его некоторым образом упрощенное представление; при упрощении мы будем «склеивать» некоторые внутренние состояния участников; могут появиться атаки на упрощенное представление, которых не было для исходного протокола (**ложноположительное срабатывание**).
- Мы можем ограничить количество рассматриваемых параллельных/вложенных сеансов (bounded model checking); это сужает пространство перебора вариантов до конечного (**ложноотрицательное срабатывание**, пропуск атак).

Возможные исходы верификации

- **Истинно положительный результат:** инструмент отработал и построил атаку, которая присуща исходному протоколу; протокол небезопасен.
- **Истинно отрицательный результат:** инструмент отработал и не смог достичь «плохого» состояния; протокол безопасен с точки зрения формальной логики.
- **Ложноположительный результат:** инструмент отработал и построил атаку, которой нет в исходном протоколе; в таком случае необходимы дальнейшие исследования (результат не определен).
- **Ложноотрицательный результат:** инструмент отработал и не смог достичь «плохого» состояния; при этом на протокол существует атака. Рассмотрим ниже подробнее.
- **Алгоритм не завершает работу:** в общем случае проблема неразрешима; инструмент может обратиться к помощи человека; употребить всю возможную память и «упасть»; обсчитывать варианты неограниченно долго и т.д.

Ложноотрицательные результаты

Откуда могут браться атаки на протоколы, которые формально верифицированы?

- **Проблемы с точки зрения теории сложности:** вероятностная природа атаки; ошибка не в логике протокола, а в теоретико-вероятностных свойствах.
- **Плохое моделирование:** неучтенные возможности противника (атаки по побочным каналам, переполнения буфера и т.д.), некорректное моделирование механизмов (моделирование уязвимой с точки зрения криптографии функции как односторонней).
- **Ограничения инструмента:** анализ относительно небольшого числа сеансов в bounded-модели [4], анализ с малым числом участников [5].
- **Ошибки реализации:** в коде самого инструмента; компиляции; выполнения машинного кода, ...

[4] J. K. Millen, «A necessarily parallel attack».

[5] K. Bhargavan и др., «Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS».

«Доказательство стойкости»

- в рамках фиксированной модели...
- в рамках некоторых математических предположений...
- для постулируемых свойств...
- в предположениях, что инструмент работает так, как заявлено.

Инструменты формальной верификации

- Теоретико-логические: ProVerif, Tamarin, Verifpal, AVISPA, Scyther, ...
- Теоретико-сложностные + автоматизированные: CryptoVerif, EasyCrypt.
- Более полно рассмотрены в М. Barbosa и др., «SoK: Computer-aided cryptography».
- В рамках данного доклада остановились на **теоретико-логических**.

Некоторые инструменты и их различия

Инструмент	ProVerif	Tamarin	Verifpal
Представление	π -исчисление	Переписывающие правила	Нотация криптогр. протоколов
База	Дизъюнкты Хорна	Переписывающие правила над мультимножествами	Custom (определяется кодом самого инструмента)
Безопасность	Невыводимость «плохой» формулы	Отсутствие «плохого» состояния в неподвижной точке	Система ограничений не имеет решения

Некоторые инструменты и их различия

Инструмент	ProVerif	Tamarin	Verifpal
Ложнополож. ¹	Возможны (упрощение протокола)	Нет	Возможны (упрощение протокола)
Ложноотриц. ¹	Нет	Нет	Да (ограниченность моделирования)
Остановка ¹	Чаще всего ²	Не гарантирована	Всегда

[1]по модулю корректно заданной спецификации протокола и корректной реализации инструмента, т.е. «в идеале»

[2]есть классы «плохих» протоколов, для которых остановка не гарантирована

Диверсификация рисков

- Разная математическая основа инструментов.
- Разный синтаксис.
- Разные реализации.
- Разные гарантии относительно ложноположительных/ложноотрицательных результатов и завершения вычислений.

Вывод: используем разное чтобы диверсифицировать риски.

Выбранные инструменты

- **Tamarin:** один из наиболее актуальных и наиболее выразительный из существующих на настоящий момент инструментов; много примеров; много историй успеха.
- **Verifpal:** самый простой и понятный инструмент относительно непосредственно использования, почти нулевой порог вхождения.

Содержание раздела

- 1 Формальная верификация в целом
- 2 **Протокол RU-Wireguard**
- 3 Полученные результаты

Wireguard

- Протокол представлен в 2017 году в работе Д. Доненфельда [6], лёг в основу VPN-решения WireGuard.
- Два участника, далее обозначаемые как **I** (Initiator, Инициатор) и **R** (Responder, Ответчик).
- Двухэтапный протокол, устанавливающий аутентифицированный и конфиденциальный канал связи (т.н. ACCE [7]): этап выработки ключа (AKE-протокол, 1.5RTT), этап защиты соединения.

[6] J. A. Donenfeld, «WireGuard: Next Generation Kernel Network Tunnel».

[7] B. Dowling, P. Rösler и J. Schwenk, «Flexible authenticated and confidential channel establishment (fACCE): Analyzing the Noise protocol framework».

Строение протокола Wireguard



Рис. 1: Общее строение протокола Wireguard; версия без Cookie

Wireguard: участники

- Участниками рассматриваемого протокола являются две равноправные стороны.
- Участник, начинающий процесс выработки ключа и отправляющий первое сообщение, в рамках протокола называется инициатором соединения (роль — Инициатор, **I**).
- Участник, отвечающий на первое сообщение, называется ответчиком (роль — Ответчик, **R**).
- Все участники равноправны: каждый участник может быть как Инициатором в каком-либо из соединений, так и Ответчиком в каком-либо другом.

Wireguard: основное предположение

Основным является следующее предположение: участник **I** знает долговременный открытый ключ участника **R**, участник **R** знает долговременный открытый ключ участника **I**.

- Вопросы безопасного предварительного распределения исходных открытых ключей сторон выходят за рамки рассмотрения данной работы.
- На практике предположение обеспечивается с помощью предварительного распределения конфигурационных файлов с открытыми ключами сторон.
- Также можно рассматривать ситуацию, в которой помимо обозначенной выше информации участники **I** и **R** знают некоторый предварительно распределенный долговременный секретный ключ psk_{IR} .
- Если между **I** и **R** предварительный ключ не распределен, то $psk_{IR} \leftarrow 0^{klen}$.

Первый этап протокола

- Первый этап должен обеспечивать ряд свойств, которые предъявляются к АKE-протоколам: секретность сеансового ключа, взаимная явная аутентификация сторон, свежесть ключей, защита от «чтения назад» и т.д.
- Формально, первый этап протокола состоит **из двух пересылок** (1-RTT).
- Фактически Ответчик **не может пересылать данные** до того, как получит первое сообщение с данными от Инициатора.
- Эта особенность связана с тем, что никакой 1-RTT-протокол не может обеспечить свойства KCI и стойкость при компрометации долговременных ключей (см. [8]).

[8]H. Krawczyk, «HMQV: A high-performance secure Diffie-Hellman protocol».

Второй этап протокола

- Второй этап работы протокола принимает на вход сеансовый ключ, выработанный в ходе работы первого этапа протокола и использует его непосредственно для защиты данных, передаваемых между участниками протокола.
- Второй этап должен обеспечивать такие свойства как конфиденциальность данных, целостность данных и присоединенных данных, защиту от повторной отправки сообщений.

Паттерн IKpsk2

Инициатор

Ответчик

```

                                <- s
                                .....
e, es, s, ss                ----->
                                <----- e, ee, se, psk
  
```

Таблица 1: Паттерн IKpsk2 семейства протоколов Noise

В любой момент времени у участника есть некоторый «накопленный» сеансовый ключ; результат очередного выполнения процедуры DH «примешивается» к накопленному ключу.

Все сообщения, передаваемые в канале, защищаются на накопленном сеансовом ключе.

Свойства безопасности: этап 1 (Э1)

- C1: взаимная (явная) аутентификация сторон;
- C8: конфиденциальность ключа;
- C9: (неявная/implicit) аутентификация ключа, в виде: **не более двух честных участников протокола получили одинаковый сеансовый ключ**;
- C10: (явная/explicit) аутентификация ключа, подтверждение владения ключом (key confirmation), в виде: **не менее двух честных участников протокола получили одинаковый сеансовый ключ**;
- C11: стойкость при компрометации производных сеансовых ключей;
- C12: стойкость при компрометации долговременных ключей (perfect forward secrecy).

[8]А. Ю. Нестеренко и А. М. Семенов, «Методика оценки безопасности криптографических протоколов».

Свойства безопасности: этап 1 (Э1)

- C13: «свежесть» ключей (свойство формирования новых ключей, key freshness);
- C14: защита от навязывания ключевых значений;
- C15: защита от навязывания параметров безопасности;
- C18: анонимность субъекта — в форме сокрытия идентификатора относительно пассивного противника («identity hiding» из работ J. A. Donenfeld, *WireGuard: Next Generation Kernel Network Tunnel. Draft revision*; B. Lipp, B. Blanchet и K. Bhargavan, «A mechanised cryptographic proof of the WireGuard virtual private network protocol»);
- C20: защита от DoS-атак;
- C22: защита от KCI-атак;
- C23: защита от UKS-атак;

Свойства безопасности: этап 2 (Э2)

- С2: аутентификация сообщений;
- С3: целостность сообщений;
- С4: защита от повторов (в рамках заданного интервала времени);
- С5: неявная аутентификация получателя;
- С8: конфиденциальность сеансовых ключей;
- С16: конфиденциальность данных;
- С17: инвариантность отправителя;
- С21: защита от атак по побочным каналам;

Упрощение свойств безопасности

- АKE-протокол, аутентификация следует из неявной + явной аутентификации ключа: $C9 + C10 \rightarrow C1, C5$.
- Конфиденциальность сеансовых ключей по умолчанию рассматриваем в условиях компометации ключей других сеансов: $C11 \rightarrow C8$.
- Считаем, что ключи привязаны к параметрам протокола через стенограмму; в таком случае стойкость к навязыванию параметров $C14, C15$ следует из «свежести» ключей $C13$ и их секретности при утечке ключей других сеансов $C11$.
- Побочные каналы и DoS-атаки не рассматриваем: исключаем $C20, C21$.

Оставшиеся свойства

- Свойства для первого этапа: C9 – C10 (аутентификация ключа), C11 (стойкость при утечке сеансовых), C12 (PFS), C13 («свежесть» ключа), C18 (анонимность), C22 (KCI), C23 (UKS).
- Свойства для второго этапа: C3 (целостность), C4 (replay), C16 (конфиденциальность).

Анонимность субъекта C18

- Более сложное свойство для формального анализа: большинство свойств связаны с индивидуальными сеансами выполнения протокола («траекториями»): т.н. trace properties; можно формализовать как identity hiding.
- В общем случае свойство анонимности говорит о неразличимости двух сеансов: т.н. observational equivalence property; т.е. работает с множествами траекторий и, вообще говоря, не может быть сформулировано как «недостижимость» некоторого плохого состояния¹ в конкретной траектории (если рассматривать сеансы по отдельности).

[1]ср. с моделью, предложенной в Е. Alekseev и S. Kyazhin, «Probing the security landscape for authenticated key establishment protocols»: атаки на протокол определяются в терминах плохих исходов отдельных сеансов, т.е. по существу являются трассе-свойствами и не в состоянии «отловить» такие свойства как (не)отказуемость или анонимность

Возможности противника

- **Канал:** $C3$, активный противник — задержка, модификация, замена, удаление, генерация сообщений в канале (модель Долева-Яо).
- **Регистрация противника:** $AR1 \cup AR2$ — регистрации противника как легитимного участника во время работы протокола; одна и та же ключевая пара используется в разных ролях.
- **Регистрация участников:** $UR1 \cup UR2$ (см. выше).
- **Взаимодействие с участниками:** $UA1 \cup UA2 \cup UA3 \cup UA4 \cup UA5 \cup UA6 \cup UA7$ — параллельные сеансы, навязывание взаимодействия двух легитимных участников, компрометация сеансовых ключей, компрометация долговременных/эфемерных закрытых и секретных ключей до/после сеансов.

[8]Е. Alekseev и S. Kyazhin, «Probing the security landscape for authenticated key establishment protocols».

Существующие результаты: Wireguard

Ист.	Инстр.	АКЕ-свойства								защ. канала		
		C9	C10	C11	C12	C13	C18	C22	C23	C3	C4	C16
[9]	Tamarin	+	+	+	+	+	id $\overline{\text{mac}}$	—	+	—	—	—
[10]	Human	+	+	+	+	+	—	+	+	—	—	—
[11]	CryptoVerif	+	+	+	+	+	id $\overline{\text{mac}}$	+	+	+	+	+
[12]	Tamarin, Proverif	+	+	+	+	+	diff mac^+	+	+	+	+	+

[9]J. A. Donenfeld и K. Milner, «Formal verification of the WireGuard protocol».

[10]B. Dowling и K. G. Paterson, «A cryptographic analysis of the WireGuard protocol».

[11]B. Lipp, «A mechanised computational analysis of the Wireguard virtual private network protocol»;
B. Lipp, B. Blanchet и K. Bhargavan, «A mechanised cryptographic proof of the WireGuard virtual private network protocol».

[12]P. Lafourcade, D. Mahmoud и S. Ruhault, «A Unified Symbolic Analysis of WireGuard».

Существующие результаты: IKpsk2

Ист.	Инстр.	AKE-свойства								защ. канала		
		C9	C10	C11	C12	C13	C18	C22	C23	C3	C4	C16
[13]	ProVerif	+	\pm	—	+	—	—	+		+	—	+
[14]	Tamarin	+	+	+	+	+	—	+	—	+	+	+
[15]	Tamarin	+	+	+	+	+	diff	+	—	+	+	+

См. также работы B. Dowling, P. Rösler и J. Schwenk, «Flexible authenticated and confidential channel establishment (fACCE): Analyzing the Noise protocol framework» с общей моделью безопасности fACCE.

[13]N. Kobeissi, G. Nicolas и K. Bhargavan, «Noise explorer: Fully automated modeling and verification for arbitrary Noise protocols».

[14]A. Suter-Dörig, «Formalizing and verifying the security protocols from the Noise framework».

[15]G. Girol и др., «A spectral analysis of noise: A comprehensive, automated, formal analysis of {Diffie-Hellman} protocols».

Важные изменения с точки зрения формального анализа

Механизм	Реализация
Согласование ключа	VKO
Группа точек Э.К.	id-tc26-gost-3410-2012-256-paramSetA id-tc26-gost-3410-2012-256-paramSetC

[1] Замена блочного шифра, функции хэширования, псевдослучайной функции, функции выработки ключа KDF, AEAD-режима не принципиальна с точки зрения формального анализа

Содержание раздела

- 1 Формальная верификация в целом
- 2 Протокол RU-Wireguard
- 3 Полученные результаты

Verifpal: пример моделирования

```
// Initializing Responder
principal Responder[
  knows private sk_r
  pk_r = G^sk_r
]
```

```
// Secure transmission
// of static key
// <- [s]
// ...
```

```
Responder -> Initiator: [pk_r]
```

Инициатор

Ответчик

```
<- s
.....
```

Verifpal: пример моделирования

```
// First Handshake message
// -> e, es, s, ss
principal Initiator[
  // initialization
  knows private sk_i
  pk_i = G^sk_i
  // protocol parameters
  knows public label
  // ephemeral keys generation
  generates esk_i
  epk_i = G^esk_i
  // es
  es_i = pk_r^esk_i
  ki_1 = HKDF(es_i, label, nil)
  static = AEAD_ENC(ki_1, pk_i, nil)
  // ss
  ss_i = pk_r^sk_i
  ki_2 = HKDF(ss_i, ki_1, nil)
]
Initiator -> Responder: epk_i, static
```

Инициатор

e, es, s, ss ----->

Ответчик

Verifpal: результаты

- C9, C10: аутентификация участника **I**, в том числе при компрометации долговременного ключа участника **I** (KCI-атаки, C22).
- C11: конфиденциальность выработанных ключей.
- C12: конфиденциальность выработанных ключей и отправленных сообщений при утечке: статических ключей участников; эфемерных ключей участников.
- C13: свежесть ключей.
- C3: целостность передаваемых сообщений.
- C16: конфиденциальность передаваемых сообщений от **I** к **R**.
- C18: ограниченно, в виде **несвязываемости** зашифрованного статического ключа *static* и передаваемого сообщения *ctxt_i*; *zero* и *ctxt_r*.

Verifpal: ложноположительные атаки

- C9, C10: ложноположительная атака на свойство аутентификации участника **R**; как следствие: ложноположительные атаки на свойство KCI для участника **I**.
- C16: конфиденциальность и целостность передаваемых сообщений от **R** \rightarrow **I**.

Verifpal: ограничения

- Нет понятия параллельных сеансов в явном виде (только при анализе) \Rightarrow невозможно выразить свойство секретности при компрометации ключей других сеансов (C11).
- Сложности со свойством UKS C23 (упирается в параллельные сеансы).
- В примитиве AEAD_ENC отсутствует *IV* — невозможно уловить replay-атаки в виде «переиспользование ключа» C4.

Tamarin: моделирование

- За основу взяли работу J. A. Donenfeld и K. Milner, «Formal verification of the WireGuard protocol».
- Заменяли все места, связанные с выработкой ключа по протоколу Диффи-Хеллмана на VKO с фиксированным значением UKM:

$$e_{sr} = pkR^{\wedge} ekI \Rightarrow$$

$$\Rightarrow vkoES = h((pkR^{\wedge}'ukm')^{\wedge} ekI).$$

- Увеличивали время на верификацию timeout.

Tamarin: результаты

- Успешная верификация свойств, доказанных в J. A. Donenfeld и K. Milner, «Formal verification of the WireGuard protocol».
- Добавление h при вычислении VKO увеличивает время вывода примерно в 6 раз.

Выводы

Объект	Инстр.	AKE-свойства								защ. канала		
		C9	C10	C11	C12	C13	C18	C22	C23	C3	C4	C16
IKpsk2	Verifpal	± I	± I	± C8	+	+	± unlink	± R	N/A	± I	N/A	± I
ruWG simpl. ¹	Tamarin	+	+	+	+	+	± id	—	+	—	—	—

[1]упрощенный протокол ruWG: нет *mac1*, *mac2*, *Cookie*, применяются некоторые упрощения при вычислении цепочек хэш-значений

Возможные дальнейшие направления исследований

- Добавление конфиденциальности, целостности передаваемых данных в анализ Tamarin.
- Аккуратное моделирование группы точек эллиптической кривой.
- Свойство анонимности в форме **diff** (observational equivalence).
- Расширение множества изученных и верифицированных свойств; более точная таксономия на основе возможностей противника.
- Учет возможности навязывания статических/эфемерных значений.
- Формальная верификация для инструментов, работающих на стыке теоретико-сложностных и теоретико-логических инструментов: CryptoVerif / EasyCrypt.

Спасибо за внимание!



tsaregorodtsev@aktiv-company.ru
sma@aktiv-company.ru



www.rutoken.ru
www.aktiv-company.ru








+7 495 925-77-90








РусКрипто







Список литературы I

-  Alekseev, E. и S. Kyazhin. «Probing the security landscape for authenticated key establishment protocols». В: *The 12th Workshop on Current Trends in Cryptology (CTCrypt 2023)*. 2023.
-  Barbosa, M. и др. «SoK: Computer-aided cryptography». В: *2021 IEEE symposium on security and privacy (SP)*. IEEE. 2021, с. 777—795.
-  Bhargavan, K. и др. «Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS». В: *2014 IEEE Symposium on Security and Privacy*. IEEE. 2014, с. 98—113.
-  Chadha, R. и др. «Automated verification of equivalence properties of cryptographic protocols». В: *ACM Transactions on Computational Logic (TOCL)* 17.4 (2016), с. 1—32.
-  Donenfeld, J. A. «WireGuard: Next Generation Kernel Network Tunnel». В: *Proceedings of the Network and Distributed System Security Symposium (NDSS'17)*. 2017, с. 1—12.



Список литературы II

-  Donenfeld, J. A. *WireGuard: Next Generation Kernel Network Tunnel. Draft revision.* <https://www.wireguard.com/papers/wireguard.pdf>. 2020.
-  Donenfeld, J. A. и K. Milner. «Formal verification of the WireGuard protocol». В: *Technical Report, Tech. Rep.* (2017).
-  Dowling, B. и K. G. Paterson. «A cryptographic analysis of the WireGuard protocol». В: *International Conference on Applied Cryptography and Network Security*. Springer. 2018, с. 3—21.
-  Dowling, B., P. Rösler и J. Schwenk. «Flexible authenticated and confidential channel establishment (fACCE): Analyzing the Noise protocol framework». В: *IACR International Conference on Public-Key Cryptography*. Springer. 2020, с. 341—373.
-  Girol, G. и др. «A spectral analysis of noise: A comprehensive, automated, formal analysis of {Diffie-Hellman} protocols». В: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, с. 1857—1874.

Список литературы III

-  Kobeissi, N., G. Nicolas и K. Bhargavan. «Noise explorer: Fully automated modeling and verification for arbitrary Noise protocols». В: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2019, с. 356—370.
-  Krawczyk, H. «HMQV: A high-performance secure Diffie-Hellman protocol». В: *Annual international cryptology conference*. Springer. 2005, с. 546—566.
-  Lafourcade, P., D. Mahmoud и S. Ruhault. «A Unified Symbolic Analysis of WireGuard». В: *Usenix Network and Distributed System Security Symposium*. 2024.
-  Lipp, B. «A mechanised computational analysis of the Wireguard virtual private network protocol». Дис. ... маг. Karlsruhe Institute of Technology, 2018.
-  Lipp, B., B. Blanchet и K. Bhargavan. «A mechanised cryptographic proof of the WireGuard virtual private network protocol». В: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2019, с. 231—246.
-  Millen, J. K. «A necessarily parallel attack». В: *Contract 30602.98-C* (1999), с. 0258.

Список литературы IV

-  Suter-Dörig, A. «Formalizing and verifying the security protocols from the Noise framework». Дис. ... маг. ETH Zurich, 2018.
-  Нестеренко, А. Ю. и А. М. Семенов. «Методика оценки безопасности криптографических протоколов». В: *Прикладная дискретная математика* 56 (2022), с. 33—82.