# Анализ информации TLS-сертификатов оконечных узлов и меток времени (SCT) для российских логов Certificate Transparency

Александр Венедюхин
Павел Якубов

**Аннотация**

Статья посвящена анализу работы российской инфраструктуры Certificate Transparency (CT). Исследуется соответствие TLS-сертификатов, обнаруженных на серверах (оконечных узлах), данным из публичных логов Certificate Transparency (CT-логов), а также проверяется валидность и корректность подписанных меток времени (SCT). Разработанное программное обеспечение выявило ряд несоответствий, включая расхождения в метках времени и отсутствие публикации сертификатов в отдельных логах. Результаты демонстрируют, что российская система CT в целом работоспособна и обеспечивает необходимую открытость, но требует повышенного внимания к устранению ошибок удостоверяющих центров для поддержания доверия.

**Ключевые слова:**

Certificate Transparency, TLS-сертификат, удостоверяющий центр, пресертификат, публичный лог, доверие, ключ, метка времени, оконечный узел, инфраструктура, Интернет.

## Введение

TLS-сертификаты казались надёжным способом защитить передаваемые данные до тех пор, пока в 2011 году не был взломан удостоверяющий центр (УЦ) DigiNotar [1]. Получив доступ к ключам удостоверяющего центра, злоумышленники смогли скомпрометировать данные пользователей Интернета, выпустив при помощи доверенных ключей данного УЦ большое количество сертификатов для имён известных веб-сервисов без ведома администраторов этих сервисов. Это позволило подменять TLS-соединения, притворившись легитимными владельцами ресурса. Ответом на подобные угрозы является система Certificate Transparency.

Certificate Transparency — группа технологий (система), предназначенная для доверенной регистрации TLS-сертификатов, выпускаемых удостоверяющими центрами, и публикации сведений об этих сертификатах. Certificate Transparency (далее — CT) была запущена для глобальной сети Интернет в 2012 году сразу после взлома DigiNotar и широко используется в настоящий момент. Предполагается, что она позволит вести независимый мониторинг деятельности удостоверяющих центров по выпуску TLS-сертификатов.

Принцип работы Certificate Transparency [2] в текущей реализации (схематично показан на рис. 1):

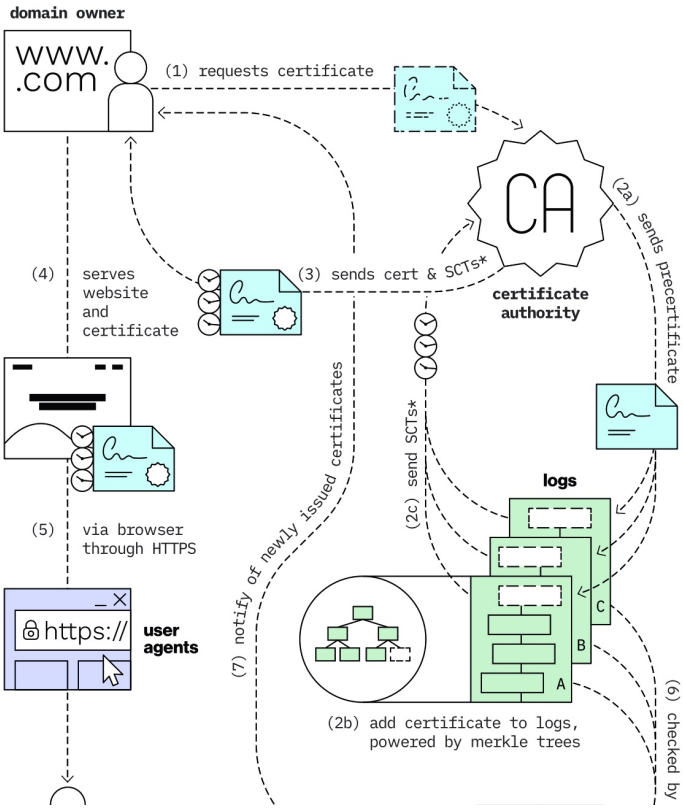


**Рис. 1. Схема работы Certificate Transparency [2].**

- пользователь запрашивает необходимый ему сертификат у удостоверяющего центра;
- вместо немедленного выпуска сертификата, как это было раньше, удостоверяющий центр сначала выпускает так называемый пресертификат, который почти идентичен стандартному сертификату;
- пресертификат отправляется в различные CT-логи — специальные журналы, хранящие сертификаты и пресертификаты;
- каждый CT-лог отправляет в ответ SCT-метку (Signed Certificate Timestamp) — структуру с отметкой времени и подписью лога, подтверждающую, что пресертификат был предъявлен оператору лога;
- удостоверяющий центр выпускает сертификат, включая в его состав полученные от оператора лога SCT-метки;
- итоговый сертификат возвращается пользователю.

Безопасность Certificate Transparency основана на том, что все выпускаемые сертификаты теперь попадают в CT-логи, а сертификаты без меток логов считаются небезопасными (не принимаются браузерами). Структура логов позволяет лишь добавлять записи, что исключает подделку данных «задним числом», а данные в них доступны всем желающим. К примеру, они могут быть использованы владельцами конкретных доменных имён и IP-адресов для выявления сертификатов, которые не должны были быть выпущены — с ошибочными данными или в результате взлома удостоверяющего центра. Благодаря CT-логам работа удостоверяющих центров становится открытой и доступной для аудита, что при регулярном мониторинге данных логов сторонними наблюдателями может предотвратить инциденты, связанные с ошибочным выпуском сертификатов.

С историей применения Certificate Transparency связан ряд достаточно громких событий в области деятельности удостоверяющих центров TLS. Например, в 2017 году данные из логов Certificate Transparency послужили доказательством того, что удостоверяющий центр Symantec выпустил много сертификатов с грубым нарушением политик, и это в итоге привело к удалению данного удостоверяющего центра из списка доверенных в браузерах. В том же году данные Certificate Transparency стали основой технического расследования в отношении удостоверяющего центра WoSign, который был замечен в выпуске сертификатов со сдвинутым в прошлое интервалом валидности, что является серьёзным нарушением общепринятых принципов работы УЦ. По этой и другим причинам корневые ключи WoSign также потеряли статус доверенных в распространённых браузерах. В 2019 году при помощи Certificate Transparency были обнаружены нарушения, допущенные удостоверяющим центром Certinomis, что также привело к отказу в доверии. В 2025 году данные Certificate Transparency позволили выявить TLS-сертификаты, ошибочно выпущенные УЦ Fina CA для IP-адреса 1.1.1.1, который соответствует глобальному сервису компании Cloudflare [11]. Данные сертификаты были также выпущены без ведома Cloudflare, однако в этом случае Certificate Transparency сработала с большой задержкой: между фактическим выпуском первого ошибочного сертификата и его обнаружением прошло более полугода (!).

Тем не менее, приведённые примеры подтверждают, что в условиях работы многих удостоверяющих центров и при наличии действенных административных инструментов влияния на их деятельность Certificate Transparency может оказывать эффективную техническую поддержку для процесса контроля [3].

Для самих владельцев доменов также есть способ использовать Certificate Transparency: время от времени они могут делать запрос к мониторам CT-логов через специальные сервисы по упрощенному API (например, через веб-интерфейс), отслеживающие изменение логов и предоставляющие информацию о сертификатах, находящихся в них, и проверять, не был ли выпущен новый сертификат на их домен [4]. А при желании владельцы доменов могут и самостоятельно мониторить непосредственно CT-логи.

Российская система Certificate Transparency, обособленная по составу логов данных и (пре-)сертификатов, появилась в 2022 году. На момент выполнения основной части данной работы (2024 год) в России были представлены три организации-оператора, занимающиеся поддержкой системы Certificate Transparency и опубликовавшие свои логи Certificate Transparency: «Яндекс», ВК и Минцифры России.

У компании «Яндекс» даже существует инструкция и описание технических требований для желающих создать свой собственный лог Certificate Transparency [5]. В случае нарушения технических требований лог удаляется из списка доверенных «Яндекс.Браузера» (этот браузер рекомендуется как поддерживающий российские доверенные TLS-сертификаты на едином портале госуслуг [6]).

В представленной работе с помощью разработанного программного обеспечения производились:

- сбор TLS-сертификатов с доступных узлов сети Интернет по именам из списка найденных в (пре-)сертификатах из логов российской системы Certificate Transparency;
- сопоставление сведений в обнаруженных сертификатах с данными из логов;
- сравнение информации из меток времени логов с данными из обнаруженных сертификатов; построение статистики, сопоставляющей сертификаты, метки времени и пресертификаты;
- поиск возможных несовпадений.

Программное обеспечение было реализовано на языке Go с использованием библиотеки с открытым исходным кодом «certificate-transparency-go» [7].

## Программная реализация

Исследованные логи используют версию 1.0 Certificate Transparency. Её спецификация опубликована в документе RFC 6962 [8]. Для получения пресертификатов из CT-логов необходимы секции спецификации 4.3 «Retrieve Latest Signed Tree Head» (используется способ получения размера лога сертификатов) и 4.6 «Retrieve Entries from Log» (используются непосредственно сертификаты и пресертификаты) (рис. 2 и 3).

```
4.3.  Retrieve Latest Signed Tree Head

   GET https://<log server>/ct/v1/get-sth

   No inputs.

   Outputs:

      tree_size:  The size of the tree, in entries, in decimal.

      timestamp:  The timestamp, in decimal.

      sha256_root_hash:  The Merkle Tree Hash of the tree, in base64.

      tree_head_signature:  A TreeHeadSignature for the above data.
```

**Рис. 2. Скриншот API RFC 6962 «Retrieve Latest Signed Tree Head».**

```
4.6.  Retrieve Entries from Log

   GET https://<log server>/ct/v1/get-entries

   Inputs:

      start:  0-based index of first entry to retrieve, in decimal.

      end:  0-based index of last entry to retrieve, in decimal.

   Outputs:

      entries:  An array of objects, each consisting of

         leaf_input:  The base64-encoded MerkleTreeLeaf structure.

         extra_data:  The base64-encoded unsigned data pertaining to the
            log entry.  In the case of an X509ChainEntry, this is the
            "certificate_chain".  In the case of a PrecertChainEntry,
            this is the whole "PrecertChainEntry".
```

**Рис. 3. Скриншот API RFC 6962 «Retrieve Entries from Log».**

Описание российской системы Certificate Transparency [9] содержит также ссылку на файл со списком российских логов, удовлетворяющих API RFC 6962, а также их открытыми ключами и идентификаторами [10].

Каждый пресертификат, найденный в логах, асинхронно обрабатывается — делаются запросы на узлы, адресуемые доменными именами, которые указаны в сертификате. Наличие асинхронной обработки данных критично для производительности по следующим причинам:

- при анализе каждого сертификата программа пытается установить соединение со всеми веб-узлами под именами из сертификата; таких имён может быть несколько десятков;
- «тайм-аут» (максимальное время ожидания ответа) подобного соединения составляет 10-20 секунд;
- в логах содержится несколько тысяч пресертификатов.

Взяв средние оценки по всем показателям и произведя несложные вычисления, можно примерно оценить время работы программы при синхронной обработке всех данных в

5·10 с·5000=250000 с≈4166 мин≈69 ч≈3 суток.

При асинхронной же обработке данных время работы программы составляет около 3-5 минут. Конечно, не все соединения будут ожидать ответа 10 секунд и не все сертификаты имеют минимум пять доменных имён, но в любом случае разница в производительности между синхронным и асинхронным вариантом программы является значительной.

Для того, чтобы проанализировать, как используется (если используется вообще) найденный в логе пресертификат, необходимо сделать HTTPS-запросы к узлам, адресуемым доменными именами, указанными в нём. Если удалось создать TLS-соединение с удалённым сервером, найденным по доменному имени, значит, хотя бы одно из доменных имён сертификата валидно в том смысле, что указывает на действующий и доступный веб-узел. На следующем шаге проверяется, используется ли на сервере конкретно этот сертификат: помимо имён, сравниваются серийные номера, найденные в сертификате сервера и в CT-логе. Если номера совпали, больше не делаем запросы по именам в поисках данного сертификата.

Если сертификат найден, начинается обработка его SCT-меток. Метки времени извлекаются из сертификата и сравниваются по значению с данными из CT-лога. Если какая-то метка времени не совпала или не найден лог, соответствующий этой метке, который должен однозначно определяться по идентификатору лога (ключу) из SCT-метки, это считается ошибкой. Ошибки специальным образом записываются. Если CT-лог, которому соответствует метка, найден, тогда, с помощью общедоступной информации о публичных ключах логов, производится попытка верификации подписи SCT-метки. Неудача верификации также считается ошибкой.

В результате получены две статистики:

- в разрезе каждого удостоверяющего центра — сколько пресертификатов, выпущенных им, используются, и у скольких из них все имена недоступны для установления TLS-соединения;
- в разрезе каждого лога Certificate Transparency — сколько SCT-меток этого лога было найдено на сертификатах с оконечных узлов.

## Результат

На рисунке 3 приведён скриншот результата работы программы без информации об ошибках, а далее — текст результата. Адрес лога «~LOG URL NOT FOUND~» означает, что найти лог, которому соответствует данная SCT-метка, не удалось. Возможно, это метки лога Certificate Transparency Минцифры России 2023 года, поскольку на момент исследования данный CT-лог был недоступен.

```
ALL:  8710 USED: 1321 (15%) ERROR DIALING: 2165 (24%) from Russian Trusted Sub CA
ALL:   106 USED:    1 ( 0%) ERROR DIALING:   97 (91%) from TCI ECDSA B1 6-160
ALL:   370 USED:   60 (16%) ERROR DIALING:  131 (35%) from TCI ECDSA B1 13-240
ALL:   133 USED:   19 (14%) ERROR DIALING:   57 (42%) from TCI ECDSA B1 10-200
SCT amount: 1359 from https://ct-agate.yandex.net/2025/
SCT amount: 1359 from https://ctlog2025.mail.ru/nca2025/
SCT amount:   38 from https://ctlog2023.mail.ru/nca2023/
SCT amount:    4 from https://ct-agate.yandex.net/2024/
SCT amount: 1284 from https://25.ctlog.digital.gov.ru/2025/
SCT amount:   38 from https://ct-agate.yandex.net/2023/
SCT amount:   37 from ~LOG URL NOT FOUND~
SCT amount:   10 from https://24.ctlog.digital.gov.ru/2024/
SCT amount:    4 from https://ctlog2024.mail.ru/nca2024/
```

**Рис. 4. Скриншот результата работы программы.**

## Выводы по результатам работы программы:

1. Количество SCT-меток, найденных в логе Минцифры России 2025 года — 1284 штуки, — не совпадает с количеством в логах «Яндекса» и ВК — 1359 штук в обоих. Как оказалось, это связано с тем, что сертификаты, выпускаемые ТЦИ («Техническим центром Интернет»), не публикуются в логах Минцифры.

2. Наибольшим процентом «ERROR DIALING» — количество пресертификатов, ни к одному из имён которых не удалось подключиться — обладает удостоверяющий центр «TCI ECDSA B1 6-160» (91%). По всей видимости, это связано с тем, что основные сертификаты, выпускаемые им, это сертификаты, используемые для тестирования. Такой вывод можно сделать ещё и потому, что большая часть из 106 выпущенных приходится на два домена: 33 на домен freetls.su и 32 на домен cateam.ru. В противовес ему выступает удостоверяющий центр «Russian Trusted Sub CA», который обладает минимальным процентом «ERROR DIALING». Скорее всего, это означает, что у данного удостоверяющего центра запрашивают сертификаты конечные пользователи для доступных узлов.

3. Среди всех найденных SCT-меток только одну не удалось верифицировать (таблица 1).

**Таблица 1. Табличная запись об SCT-метке, которую не удалось верифицировать**

| SerialNumber | Common-Name | NotBe-fore | Issuer | Error | SCT-Times-tamp | Log-Times-tamp | LogURL | LogIndex | LogEn-tries | Raw-Base64 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0EE83DE68919B11E-FA5400242AC120002 | s.autoup-date.info | 24/10/2024 00:07:18 | Russian Trusted Sub CA | SCT is not verifying: failed to verify ECDSA signature | 24/10/2024 03:07:18 | 25/10/2024 00:32:01 | https:// ct-agate. yandex. net/2025/ | 4993 | [https:// ct-agate. yandex. net/2025/] | |
| 0EE83DE68919B11E-FA5400242AC120002 | s.autoup-date.info | 24/10/2024 00:07:18 | Russian Trusted Sub CA | Timestamp is wrong | 24/10/2024 03:07:18 | 25/10/2024 00:32:01 | https:// ct-agate. yandex. net/2025/ | 4993 | [https:// ct-agate. yandex. net/2025/] | |

Видно, что при анализе данного сертификата произошла также и ошибка «Timestamp is wrong». И действительно, при ближайшем рассмотрении оказалось, что у этого пресертификата и соответствующего ему сертификата время начала действия различается на секунду (таблица 2).

**Таблица 2. Табличная запись о сертификате, SCT-метку которого не удалось верифицировать, и соответствующем ему пресертификате**

| SerialNumber | Common-Name | NotBe-fore | Issuer | CanDial | Used | SCTError | SAN | LogEntries | LogEn-tries | Raw-Base64 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0EE83DE68919B11E-FA5400242AC120002 | s.autoup-date.info | 24/10/2024 00:07:17 | Russian Trusted Sub CA | TRUE | TRUE | TRUE | [s.autoup-date.info] | [https://ctlog2025.mail. ru/nca2025/ https://25. ctlog.digital.gov. ru/2025/ https://ct-ag-ate.yandex.net/2025/] | [https:// ct-agate. yandex. net/2025/] | |
| 0EE83DE68919B11E-FA5400242AC120002 | s.autoup-date.info | 24/10/2024 00:07:18 | Russian Trusted Sub CA | TRUE | TRUE | TRUE | [s.autoup-date.info] | [https://ct-agate.yan-dex.net/2025/] | [https:// ct-agate. yandex. net/2025/] | |

Такой состав сертификата свидетельствует о том, что при его формировании удостоверяющим центром была допущена грубая ошибка, поскольку несовпадение времён начала действия сертификата и соответствующего ему пресертификата вызывает сомнения относительно подлинности сертификата с SCT-меткой и соответствия требованиям безопасности. Такой сертификат не может использоваться при условии контроля SCT-меток ввиду того, что при валидации SCT-метки стандартными методами воспользоваться будет нельзя, так как не совпадёт значение цифровой подписи из состава метки.

## Заключение

Текущее состояние российской системы Certificate Transparency позволяет выявлять ошибки в SCT-метках сертификатов, которые играют ключевую роль в обеспечении безопасности системы Web PKI (англ. Public Key Infrastructure — инфраструктура открытых ключей). Следует надеяться, что количество подобных проблем будет в дальнейшем минимизировано. Несмотря на найденное в ходе исследования несоответствие сертификата пресертификату, российская система Certificate Transparency является полностью рабочей, удовлетворяет принятому API, обеспечивает открытость данных и доступна любому внешнему наблюдателю, что является положительным результатом.

## Список литературы:

[1] Xakep. Подробности взлома DigiNotar. URL: https://xakep.ru/2012/11/01/59572/ (дата обращения: 20.12.2024).

[2] certificate.transparency.dev. Принцип работы Certificate Transparency. URL: https://certificate.transparency.dev/howctworks/ (дата обращения: 20.12.2024).

[3] ТЦИ. Certificate Transparency в современном интернете. URL: https://www.tcinet.ru/press-centre/articles/7751/?ysclid=m53mxtf8vn349936923 (дата обращения: 20.12.2024).

[4] HackWare. Мониторинг появления новых субдоменов в реальном времени. URL: https://hackware.ru/?p=16188 (дата обращения: 20.12.2024).

[5] Яндекс. Политика доверия Certificate Transparency log. URL: https://yandex.ru/support/browser-beta/ru/security/policy-ct-log.html?ysclid=m53nq6wlgd783023557 (дата обращения: 20.12.2024).

[6] Госуслуги. Поддержка работы сайтов с российскими сертификатами. URL: https://www.gosuslugi.ru/crt (дата обращения: 20.12.2024).

[7] GitHub. Исходный код и описание библиотеки «certificate-transparency-go». URL: https://github.com/google/certificate-transparency-go (дата обращения: 20.12.2024).

[8] IETF Datatracker. RFC 6962 — Certificate Transparency. URL: https://datatracker.ietf.org/doc/html/rfc6962 (дата обращения: 20.12.2024).

[9] Habr. Статья 2022 года с описанием устройства российской системы CT. URL: https://habr.com/ru/companies/yandex/articles/667300/ (дата обращения: 20.12.2024).

[10] Ссылка на список логов Certificate Transparency. URL: https://browser-resources.s3.yandex.net/ctlog/ctlog.json (дата обращения: 20.12.2024).

[11] Addressing the unauthorized issuance of multiple TLS certificates for 1.1.1.1. URL: https://blog.cloudflare.com/unauthorized-issuance-of-certificates-for-1-1-1-1/ (дата обращения: 16.09.2025).

## Об авторах

Венедюхин Александр Анатольевич,
ведущий специалист ФРСТ «ИнДата»

Павел Якубов,
студент МГТУ им. Н.Э. Баумана