

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ,
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004

DOI: 10.17212/2782-2230-2026-3-24-40

**ЗАЩИТА КАНАЛОВ СВЯЗИ ЦИФРОВЫХ ДВОЙНИКОВ
ОТ АТАК В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ
СИСТЕМАХ***

Н.С. СИЛЬЧЕНКО¹, Р.А. ПЕРМЯКОВ²

¹ РФ, 630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, лаборант кафедры защиты информации. E-mail: silchenko.2022@stud.nstu.ru

² РФ, 630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, лаборант кафедры защиты информации E-mail: permyakov@corp.nstu.ru

Технология цифровых двойников активно внедряется в промышленный интернет вещей для мониторинга и управления сложными киберфизическими системами в режиме реального времени. Двусторонний характер обмена данными между физическим объектом и его виртуальной копией порождает дополнительные векторы атак, причем особую опасность представляют атаки на каналы связи, использующие легковесный протокол MQTT. В статье разработана модель угроз безопасности информации для каналов связи цифровых двойников в соответствии с методикой ФСТЭК России, выделено шесть актуальных угроз, включая подмену данных, повторное воспроизведение и отказ в обслуживании. Создан экспериментальный стенд на базе эмулятора датчика температуры, MQTT-брокера HiveMQ и платформы цифровых двойников Eclipse Ditto, развернутых в Docker-контейнерах. Реализованы сценарии атак MITM с подменой данных, Replay-атаки и DoS-атаки. Показано, что незащищенный MQTT-канал уязвим для всех перечисленных угроз. Предложен комплекс механизмов защиты: TLS-шифрование канала, проверка временных меток в JavaScript-маппере и правила iptables для ограничения интенсивности трафика. Экспериментально подтверждена эффективность предложенных мер: MITM-атака становится невозможной при корректной проверке сертификата, Replay-атака предотвращается для рассматриваемого сценария отбрасыванием устаревших сообщений, DoS-атака существенно снижается за счет фильтрации на уровне межсетевого экрана.

Ключевые слова: цифровой двойник, MQTT, TLS, MITM, Replay-атака, DoS-атака, iptables, Eclipse Ditto, HiveMQ, распределенные вычислительные системы, модель угроз

* Статья получена 02 июля 2026 г.

ВВЕДЕНИЕ

В условиях цифровой трансформации промышленности и перехода к концепции Industry 4.0 технология цифровых двойников (ЦД) становится ключевым инструментом для мониторинга, анализа и управления сложными техническими системами в реальном времени. Цифровой двойник представляет собой виртуальную копию физического объекта, синхронизированную с ним по двунаправленным каналам связи [1, 2]. Эта синхронизация позволяет не только наблюдать за состоянием объекта, но и прогнозировать его поведение, выявлять аномалии и даже формировать управляющие воздействия.

Однако именно двусторонний характер обмена данными существенно расширяет поверхность атаки. Злоумышленник, получивший доступ к каналу связи, может не только перехватывать конфиденциальную телеметрию, но и модифицировать ее в реальном времени, повторно воспроизводить легитимные сообщения или инициировать отказ в обслуживании [3, 4]. Особую опасность представляет протокол MQTT, широко применяемый в промышленном интернете вещей благодаря своей легкости и поддержке модели публикации/подписки. Стандартная конфигурация MQTT не предусматривает встроенного шифрования, аутентификации или контроля целостности, что делает протокол уязвимым для атак типа «человек посередине», повторного воспроизведения и отказа в обслуживании [5, 6].

Существующие подходы к обеспечению информационной безопасности, например построение центров мониторинга и реагирования на инциденты (SOC), ориентированы на обнаружение уже известных паттернов атак и часто являются реактивными. Цифровой двойник, напротив, благодаря своей способности моделировать будущие состояния системы может реализовать проактивную парадигму защиты, выявляя аномалии на ранних стадиях [7]. Однако для этого сам канал связи, по которому передаются данные, должен быть надежно защищен.

Цель настоящей работы – исследование и экспериментальная оценка механизмов защиты каналов связи цифровых двойников от атак в распределенных вычислительных системах. В работе решаются следующие задачи: анализ архитектуры каналов связи и разработка модели угроз по методике ФСТЭК России; создание экспериментального стенда, эмулирующего работу цифрового двойника на базе протокола MQTT; реализация сценариев типовых атак; внедрение комплекса защитных механизмов и оценка их эффективности, включая влияние на временные характеристики передачи данных.

1. АРХИТЕКТУРА КАНАЛОВ СВЯЗИ ЦИФРОВЫХ ДВОЙНИКОВ И МОДЕЛЬ УГРОЗ

Ключевой особенностью цифрового двойника является двусторонняя автоматическая связь с физическим объектом. По каналу «объект → двойник» передаются показания датчиков, параметры работы оборудования, журналы событий. По обратному каналу «двойник → объект» могут отправляться команды управления, рекомендации, аварийные сигналы [1, 8].

В качестве транспорта в распределенных системах часто используется протокол MQTT. Он работает поверх TCP, поддерживает три уровня качества обслуживания (QoS 0,1,2) и является легковесным, что важно для устройств с ограниченными ресурсами. Однако стандартная реализация MQTT не содержит средств шифрования, аутентификации и контроля целостности сообщений.

На основе методики ФСТЭК России [9] и банка данных угроз [10] была разработана модель угроз для экспериментального стенда, включающего эмулятор датчика, MQTT-брокер и платформу цифровых двойников. Выделены следующие актуальные угрозы (табл. 1).

Таблица 1

Table 1

Актуальные угрозы безопасности информации

Current threats to information security

Идентификатор	Наименование угрозы	Краткое описание
УБИ.069	Неправомерные действия в каналах связи	MITM-атака с подменой значения температуры
УБИ.006	Внедрение кода или данных	Инъекция ложных телеметрических сообщений
УБИ.083	Несанкционированный доступ	Подключение к MQTT-брокеру без аутентификации
УБИ.140	Отказ в обслуживании	DoS-атака путём флуда сообщениями
УБИ.192	Использование уязвимых версий ПО	Эксплуатация известных уязвимостей в HiveMQ, Ditto
УБИ.116	Перехват данных в сети	Сниффинг MQTT-трафика

Основными уязвимостями стенда являются: передача данных в открытом виде, отсутствие аутентификации, отсутствие проверки целостности и свежести сообщений, а также отсутствие rate limiting на уровне брокера.

2. ЭКСПЕРИМЕНТАЛЬНЫЙ СТЕНД

Для экспериментального исследования был разработан стенд на базе двух виртуальных машин с ОС Ubuntu 22.04 LTS и одной машины с ОС Windows 10. Серверные компоненты (MQTT-брокер HiveMQ и платформа Eclipse Ditto) развернуты в Docker-контейнерах (табл. 2).

Т а б л и ц а 2
T a b l e 2

Конфигурация узлов стенда
Configuration of the stand nodes

Роль узла	ОС	CPU	RAM	IP-адрес
Сервер приложений (брокер + Ditto)	Ubuntu 22.04 LTS	4 vCPU	8 GB	192.168.0.156
Атакующая машина (MITM, DoS, Replay)	Ubuntu 22.04 LTS	2 vCPU	4 GB	192.168.0.192
Эмулятор датчика	Windows 10 Pro	4 vCPU	4 GB	192.168.0.155

Эмулятор датчика температуры написан на Python с использованием библиотеки paho-mqtt. Каждое сообщение содержит JSON-структуру, представленную на рис. 1.

```
{
  "sensor_id": "temp-sensor-91",
  "temperature": 20.1,
  "timestamp": 1776144037.2124453
}
```

Рис. 1. JSON-структура телеметрического сообщения от сенсора

Fig. 1. JSON structure of the telemetry message from the sensor

Цифровой двойник создан в Eclipse Ditto как Thing sensor:temp1 с feature temperature, содержащей свойства value и timestamp. Интеграция с MQTT реализована через встроенный MQTT-коннектор, который подписывается на топик sensors/temperature. Для преобразования входящих сообщений во внутренний протокол Ditto используется JavaScript-маппер. Такая архитектура обеспечивает гибкость экспериментальной среды и позволяет в реальном времени отслеживать изменения состояния цифрового двойника при поступлении телеметрии.

Схема взаимодействия компонентов представлена на рис. 2 и включает следующие потоки:

- штатная передача данных от эмулятора к брокеру (порт 1883 без шифрования, порт 8883 – с TLS);
- в режиме MITM-атаки трафик перенаправляется на прокси (порт 1884), который модифицирует значение температуры;
- DoS-атака генерирует многопоточный флуд сообщениями;
- Replay-атака состоит из фазы sniffинга (запись сообщений) и фазы воспроизведения.

Такое разделение потоков позволяет последовательно эмулировать каждый тип угроз и оценивать эффективность соответствующих защитных механизмов в контролируемых условиях.

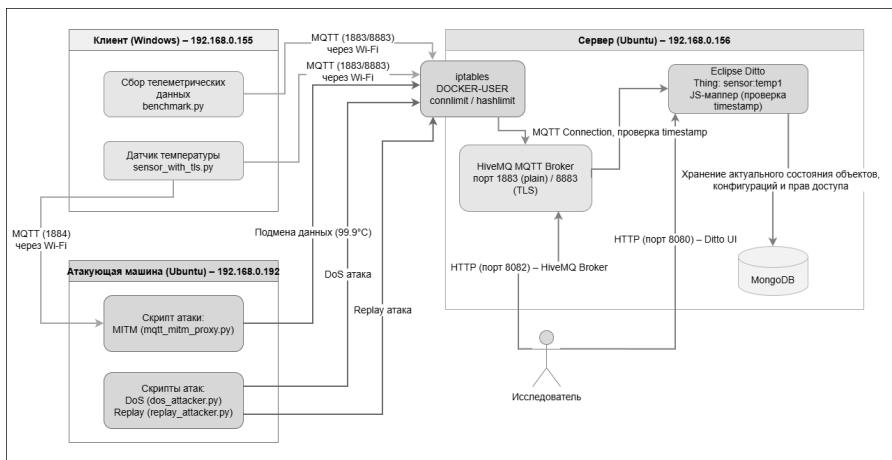


Рис. 2. Схема взаимодействия компонентов

Fig. 2. Component interaction diagram

3. МОДЕЛИРОВАНИЕ АТАК

3.1. MITM-АТАКА С ПОДМЕНОЙ ДАННЫХ

Разработан прокси-сервер на Python, который перехватывает TCP-соединение между эмулятором и брокером. Прокси-сервер анализирует проходящие MQTT-сообщения, находит в JSON-полезной нагрузке поле temperature и заменяет его значение на 99,9 °C. В незащищенном канале (порт 1883) атака успешна: цифровой двойник отображает ложное значение температуры. При этом в логах прокси-сервера фиксируются сообщения MODIFIED (рис. 3). Успех атаки ожидаемый и объясняется отсутствием шифрования. Однако важно отметить, что даже при TLS атака MITM возможна, если клиент не проверяет сертификат, – это ограничение следует учитывать.

Incoming Thing Updates

Reset

Add basic filter...

Revision	Action	Path	Field(s)	Modified
8965	modified	/features/temperature/propert...		2026-04-23 05:16:25 434254980
8966	modified	/features/temperature/propert...		2026-04-23 05:21:28 399148924
8967	modified	/features/temperature/propert...		2026-04-23 05:21:30 381128104
8968	modified	/features/temperature/propert...		2026-04-23 05:21:32 384150788
8969	modified	/features/temperature/propert...		2026-04-23 05:21:34 387787480
8970	modified	/features/temperature/propert...		2026-04-23 05:21:36 387565338
8971	modified	/features/temperature/propert...		2026-04-23 05:21:38 389997930
8972	modified	/features/temperature/propert...		2026-04-23 05:21:40 391636118
8973	modified	/features/temperature/propert...		2026-04-23 05:21:42 395070223
8974	modified	/features/temperature/propert...		2026-04-23 05:21:44 394542500
8975	modified	/features/temperature/propert...		2026-04-23 05:21:46 397802815
8976	modified	/features/temperature/propert...		2026-04-23 05:23:03 117528430
8977	modified	/features/temperature/propert...		2026-04-23 05:23:05 116058344
8978	modified	/features/temperature/propert...		2026-04-23 05:23:07 117034078
8979	modified	/features/temperature/propert...		2026-04-23 05:23:09 119637134
8980	modified	/features/temperature/propert...		2026-04-23 05:23:11 121482682

Thing Update Detail

Show Only context

```
{
  "revision": 8980,
  "_created": "2026-04-23T05:02:59.713091736Z",
  "_modified": "2026-04-23T05:23:11.121482682Z",
  "_context": {
    "topic": "sensor/temp/things/twz/events/modified",
    "path": "/features/temperature/properties",
    "value": {
      "value": 99.9,
      "timestamp": 1776921791.045335
    }
  }
}
```

Рис. 3. Измененные данные, полученные цифровым двойником

Fig. 3. Modified data received by the digital twin

3.2. REPLAY-АТАКА

Скрипт sniffера подписывается на топик sensors/temperature и сохраняет все сообщения в файл captured_messages.json. Затем скрипт воспроизведения читает файл и публикует сохраненные сообщения в тот же топик с заданной задержкой. Поскольку платформа Ditto не проверяет временные метки, цифровой двойник обновляется устаревшими данными, теряя синхронизацию с реальным объектом (рис. 4). Повторяющиеся временные метки и значения температуры явно видны в интерфейсе Ditto (рис. 5). Уязвимость возникает из-за отсутствия проверки свежести на уровне приложения. TLS не защищает от повторов, поэтому требуется дополнительная логика на стороне двойника.

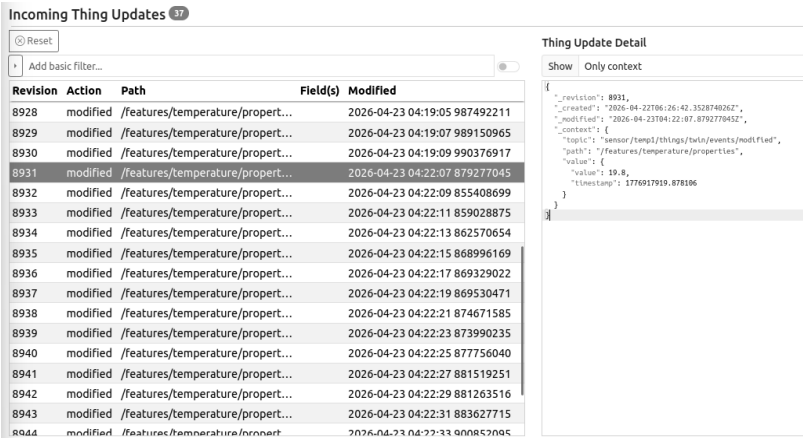


Рис. 4. Данные, полученные цифровым двойником в результате Replay-атаки, первый повтор

Fig. 4. The data obtained by the digital double as a result of the Replay attack, the first replay

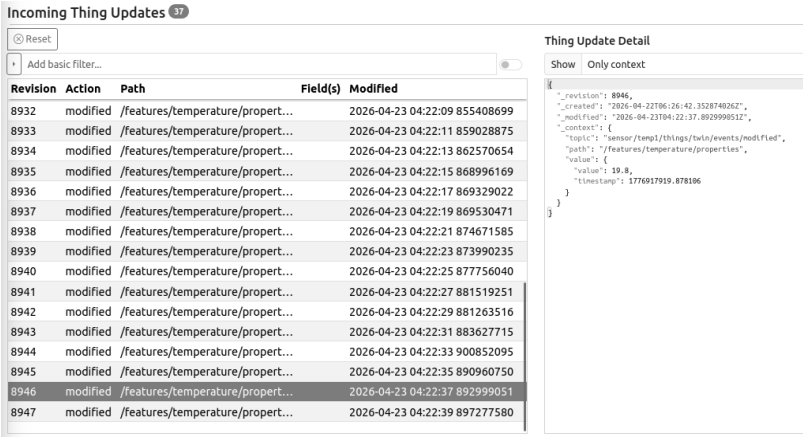


Рис. 5. Данные, полученные цифровым двойником в результате Replay-атаки, второй повтор

Fig. 5. The data obtained by the digital double as a result of the Replay attack, the second replay

3.3. DOS-АТАКА

Многопоточный скрипт (2 потока по 50 сообщений в секунду) публикует фальшивые сообщения в топик sensors/temperature. При атаке на незащищенный канал загрузка CPU контейнера connectivity Eclipse Ditto возрастает до 112 %, контейнера things – до 72 %, легитимные сообщения от реального датчика доставляются с критическими задержками или теряются. Это подтверждает уязвимость брокера и платформы к атакам типа «отказ в обслуживании» даже при сравнительно невысокой интенсивности (100 сообщений в секунду). Рост CPU до 112 % обусловлен парсингом JSON и работой маппера. Атака с интенсивностью 100 сообщений в секунду уже критична. Значит, без фильтрации на сетевом уровне платформа не выдерживает даже умеренного флуда (рис. 6, 7).

```
root@mqtt-server:~/ditto/deployment/docker# docker stats --no-stream
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O
PIDS						
abb34114c1ce	docker_nginx_1	0.04%	4.965MiB / 32MiB	15.52%	8.24MB / 8.02MB	53.2kB / 12.3kB
5						
d1c883206faf	docker_things_1	18.16%	484.2MiB / 512MiB	94.57%	162MB / 151MB	53.1MB / 24.3MB
104						
85cb2ac21b0b	docker_connectivity_1	12.18%	565.4MiB / 768MiB	73.62%	171MB / 198MB	15.6MB / 24.1MB
104						
8d0516b0c887	docker_gateway_1	8.48%	459.4MiB / 512MiB	89.72%	169MB / 173MB	51.9MB / 27.8MB
52						
4f7a11ddb10b	docker_things-search_1	6.66%	438.9MiB / 512MiB	85.71%	106MB / 96.4MB	52.5MB / 29.3MB
75						
bocddf4ac7e7	docker_swagger-ui_1	0.00%	17.45MiB / 32MiB	54.54%	13.2kB / 126B	15.9MB / 1.13MB
5						
d4436cfb8fb0	hivemq	2.34%	709.2MiB / 7.71GiB	8.98%	1.13MB / 1.24MB	2.01MB / 75.5MB
136						
6adae0b349b4	docker_ditto-ui_1	0.00%	4.723MiB / 32MiB	14.76%	21.7kB / 3.03kB	0B / 8.19kB
5						
bed2fada5db7	docker_mongodb_1	1.60%	101.5MiB / 256MiB	39.65%	16.5MB / 31MB	32.8kB / 135MB
107						
3f312abb8a3a	docker_policies_1	5.26%	450.2MiB / 512MiB	87.93%	137MB / 115MB	86MB / 37.7MB
84						

Рис. 6. Проверка нагрузки на сервере с запущенным датчиком

Fig. 6. Checking the load on a server with a running sensor

```
root@mqtt-server:~/ditto/deployment/docker# docker stats --no-stream
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O
PIDS						
abb34114c1ce	docker_nginx_1	1.51%	4.965MiB / 32MiB	15.52%	5.66MB / 5.51MB	53.2kB / 12.3kB
5						
d1c883206faf	docker_things_1	112.77%	489.6MiB / 512MiB	95.62%	87.2MB / 88.2MB	53.1MB / 17.9MB
112						
85cb2ac21b0b	docker_connectivity_1	72.54%	585.9MiB / 768MiB	76.28%	91.5MB / 106MB	15.6MB / 19.9MB
112						
8d0516b0c887	docker_gateway_1	23.83%	459.2MiB / 512MiB	89.69%	90.9MB / 90.9MB	51.9MB / 11.9MB
61						
4f7a11ddb10b	docker_things-search_1	16.53%	457MiB / 512MiB	89.26%	57.1MB / 49.8MB	52.5MB / 21.6MB
75						
bocddf4ac7e7	docker_swagger-ui_1	0.00%	17.45MiB / 32MiB	54.54%	11.9kB / 126B	15.9MB / 1.13MB
5						
d4436cfb8fb0	hivemq	2.92%	711.1MiB / 7.71GiB	9.01%	875kB / 986kB	2.01MB / 74.8MB
140						
6adae0b349b4	docker_ditto-ui_1	0.00%	4.723MiB / 32MiB	14.76%	20.4kB / 3.03kB	0B / 8.19kB
5						
bed2fada5db7	docker_mongodb_1	11.58%	103.6MiB / 256MiB	40.48%	10.7MB / 16.4MB	32.8kB / 76.2MB
107						
3f312abb8a3a	docker_policies_1	5.18%	455.6MiB / 512MiB	88.98%	71.2MB / 59.6MB	86MB / 22.6MB
82						

```
root@mqtt-server:~/ditto/deployment/docker#
```

Рис. 7. Проверка нагрузки на сервере в состоянии идущей DoS-атаки и работающего датчика

Fig. 7. Checking the load on the server in the state of an ongoing DoS attack and a working sensor

4. МЕХАНИЗМЫ ЗАЩИТЫ

4.1. ШИФРОВАНИЕ TLS ДЛЯ MQTT-КАНАЛА

Для защиты канала от перехвата и подмены данных настроен MQTT over TLS на порте 8883. С помощью утилиты `keytool` сгенерирован самоподписанный сертификат сервера, который импортирован в хранилище ключей `NiveMQ`. Эмулятор датчика модифицирован для поддержки TLS с проверкой сертификата. При попытке провести MITM-атаку прокси-сервер получает зашифрованный трафик и не может расшифровать содержимое; модификация данных становится невозможной. Захват трафика Wireshark показывает нечитаемый бинарный поток (рис. 8, 9).

No.	Time	Source	Destination	Protocol	Length	Info
1802	0.057430037	192.168.0.155	192.168.0.192	TCP	60	52343 → 1884 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
1803	0.057384937	192.168.0.192	192.168.0.155	TCP	60	1884 → 52343 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
1804	0.05530991	192.168.0.155	192.168.0.192	TCP	60	52343 → 1884 [ACK] Seq=1 Ack=1 Win=131328 Len=0
1805	0.056491296	192.168.0.155	192.168.0.192	TLSv1.2	216	Client Hello
1806	0.056474896	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=1 Ack=163 Win=64128 Len=0
1814	0.057044812	192.168.0.192	192.168.0.155	TLSv1.2	1350	Server Hello, Certificate, Server Key Exchange, Server Hello Done
1815	0.055376156	192.168.0.155	192.168.0.192	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1818	0.056432628	192.168.0.192	192.168.0.155	TLSv1.2	1276	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
1819	0.054182043	192.168.0.155	192.168.0.192	TLSv1.2	113	Application Data
1822	0.054189406	192.168.0.192	192.168.0.155	TLSv1.2	87	Application Data
1826	0.052267511	192.168.0.155	192.168.0.192	TCP	60	52343 → 1884 [ACK] Seq=315 Ack=2554 Win=131328 Len=0
1834	0.054220995	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1837	0.056524996	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=469 Win=64128 Len=0
1839	0.053847926	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1860	0.053936395	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=623 Win=64128 Len=0
1869	0.053771264	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1889	0.053764063	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=777 Win=64128 Len=0
1924	0.053490612	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1925	0.053492411	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=931 Win=64128 Len=0
1955	0.053822263	192.168.0.155	192.168.0.192	TLSv1.2	207	Application Data
1956	0.053758202	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=1084 Win=64128 Len=0
1959	0.053435019	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1987	0.053427509	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=1238 Win=64128 Len=0
1990	0.053192059	192.168.0.155	192.168.0.192	TLSv1.2	208	Application Data
1999	0.053170505	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=1392 Win=64128 Len=0
2016	0.051347476	192.168.0.155	192.168.0.192	TLSv1.2	85	Application Data
2018	0.053192059	192.168.0.155	192.168.0.192	TCP	54	1884 → 52343 [ACK] Seq=2554 Ack=1424 Win=64128 Len=0
2019	0.051567414	192.168.0.155	192.168.0.192	TCP	60	52343 → 1884 [FIN, ACK] Seq=1423 Ack=2554 Win=131328 Len=0
2020	0.051525139	192.168.0.192	192.168.0.155	TCP	54	1884 → 52343 [FIN, ACK] Seq=2554 Ack=1424 Win=64128 Len=0
2027	0.050431591	192.168.0.155	192.168.0.192	TCP	60	52343 → 1884 [ACK] Seq=1424 Ack=2555 Win=131328 Len=0

Рис. 8. Захват зашифрованного трафика MQTT over TLS

Fig. 8. Capture encrypted MQTT over TLS traffic



Рис. 9. Зашифрованный TLS-трафик

Fig. 9. Encrypted TLS traffic

4.2. ЗАЩИТА ОТ REPLAY-АТАК

В JavaScript-маппере Eclipse Ditto добавлена проверка временной метки входящего сообщения. Если timestamp отличается от текущего времени более чем на 5 секунд, сообщение отбрасывается (функция возвращает null). После внедрения этой проверки все сообщения, воспроизводимые скриптом Replay-атаки, игнорируются. Легитимные сообщения от работающего датчика продолжают обрабатываться штатно. Таким образом, достигается защита от повторного воспроизведения без модификации клиентского ПО.

Incoming JavaScript Mapping

```
1 function mapToDittoProtocolMsg(headers, textPayload, bytePay
2   var json = JSON.parse(textPayload);
3   var currentTime = Math.floor(Date.now() / 1000);
4
5   if (currentTime - json.timestamp > 5) {
6     return null;
7   }
8
9   return Ditto.buildDittoProtocolMsg(
10     'sensor',           // namespace
11     'temp1',            // id
12     'things',
13     'twin',
14     'commands',
15     'modify',
16     '/features/temperature/properties',
17     headers,
18     {
19       value: json.temperature,
20       timestamp: json.timestamp
21     }
22   );
23 }
```

Рис. 10. Модифицированный JavaScript-маппер с проверкой актуальности временной метки

Fig. 10. Modified JavaScript mapper with timestamp validity check

4.3. ЗАЩИТА ОТ DOS-АТАК

На уровне межсетевого экрана (iptables) в цепочке DOCKER-USER применены следующие правила:

- ограничение количества одновременных соединений с одного IP-адреса: `-m connlimit --connlimit-above 1 --connlimit-mask 32 -j DROP`;
- отслеживание подозрительных IP-адресов с помощью модуля recent и блокировка при превышении трех пакетов за 1 секунду;

- ограничение скорости пакетов: `-m hashlimit --hashlimit 3/sec --hashlimit-burst 6 --hashlimit-mode srcip --hashlimit-name dos_limit -j АСCEPT`.

После применения правил атакующий трафик отфильтровывается, а легитимные сообщения датчика проходят без задержек. Загрузка CPU контейнеров возвращается к штатному уровню.

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА И ОЦЕНКА ЗАДЕРЖЕК

5.1. ЭФФЕКТИВНОСТЬ ЗАЩИТЫ

Сравнительный анализ сценариев защиты показал, что ни один из механизмов по отдельности не обеспечивает полной защиты. Только их комплексное применение (TLS + проверка временных меток + iptables) снижает риски для всех трех типов атак (табл. 3).

Таблица 3

Table 3

Сравнительный анализ сценариев защиты

Comparative analysis of protection scenarios

Сценарий	MITM-атака успешна	Replay-атака успешна	DoS-атака успешна
Без защиты	Да	Да	Да
Только TLS	Нет для простого MITM в канале	Частично, зависит от модели атакующего	Да
Только защита от Replay	Да	Нет для устаревших сообщений	Да
Только защита от DoS	Да	Да	Снижает успешность
Комплексная защита	Снижает риск	Снижает риск	Снижает риск

5.2. ВЛИЯНИЕ НА ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ

Для количественной оценки накладных расходов были измерены задержки MQTT publish RTT (Round-Trip Time) в четырех режимах:

- 1) прямое соединение без TLS (базовый режим);

- 2) через MITM-прокси без TLS;
 - 3) прямое соединение с TLS;
 - 4) через MITM-прокси с TLS.
- В каждом режиме 5 серий по 100 измерений (табл. 4).

Таблица 4

Table 4

Статистические характеристики задержек MQTT publish RTT

Statistical characteristics of MQTT delays publish RTT

Сценарий	n	Mean	Median	Std	Min	Max	Q1	Q3	IQR	p90	p95	p99	Выбросы	Выбросы, %	p95/Median
Напрямую без TLS	500	8.53	8.64	3.06	2.60	15.63	5.80	11.04	5.24	12.66	13.21	14.18	0	0.00	1.53
MITM без TLS	500	9.27	9.42	3.04	3.38	17.27	6.65	11.76	5.11	13.51	14.05	14.93	0	0.00	1.49
Напрямую с TLS	500	9.30	9.29	3.12	3.34	21.67	6.64	11.86	5.22	13.28	13.87	15.23	1	0.20	1.49
MITM с TLS	500	10.44	10.65	2.98	4.18	18.48	8.02	12.58	4.57	14.39	15.11	16.54	0	0.00	1.42

Для проверки статистически значимых различий между распределениями задержек MQTT publish RTT во всех четырех экспериментальных сценариях использовался непараметрический критерий Kruskal – Wallis. Для выполнения попарных сравнений между отдельными сценариями применялся критерий Mann – Whitney U с поправкой Holm – Bonferroni на множественные сравнения. Статистический анализ (критерий Краскела – Уоллиса, $p < 0,001$) подтвердил значимые различия между распределениями (табл. 5).

Таблица 5

Table 5

Результаты статистических тестов

Statistical test results

Сравнение	Тест	Статистика	p-value	Вывод
MITM без TLS vs Напрямую без TLS	Mann-Whitney U	$U = 108488.5, Z = -3.62$	< 0.001	Значимо
Напрямую с TLS vs Напрямую без TLS	Mann-Whitney U	$U = 108131.0, Z = -3.69$	< 0.001	Значимо
MITM с TLS vs Напрямую с TLS	Mann-Whitney U	$U = 99348.0, Z = -5.62$	< 0.001	Значимо
MITM с TLS vs MITM без TLS	Mann-Whitney U	$U = 98779.0, Z = -5.74$	< 0.001	Значимо
MITM с TLS vs Напрямую без TLS	Mann-Whitney U	$U = 83585.5, Z = -9.07$	< 0.001	Значимо
Межгрупповое сравнение (все 4 сценария)	Kruskal-Wallis	$H = 86.77, df = 3$	< 0.001	Распределения различны
Сравнение дисперсий	Levene	$W = 1.10, df1 = 3, df2 = 1996$	0.347	Дисперсии равны

Использование TLS увеличивает среднюю задержку на 9 % относительно прямого незащищенного канала, а совместное применение MITM-прокси и TLS – на 23 %. При этом форма распределения сохраняется, доля аномальных выбросов не возрастает, что говорит о стабильности соединения. Полученные задержки остаются приемлемыми для систем мягкого реального времени и не нарушают работоспособность цифрового двойника (рис. 11–13).

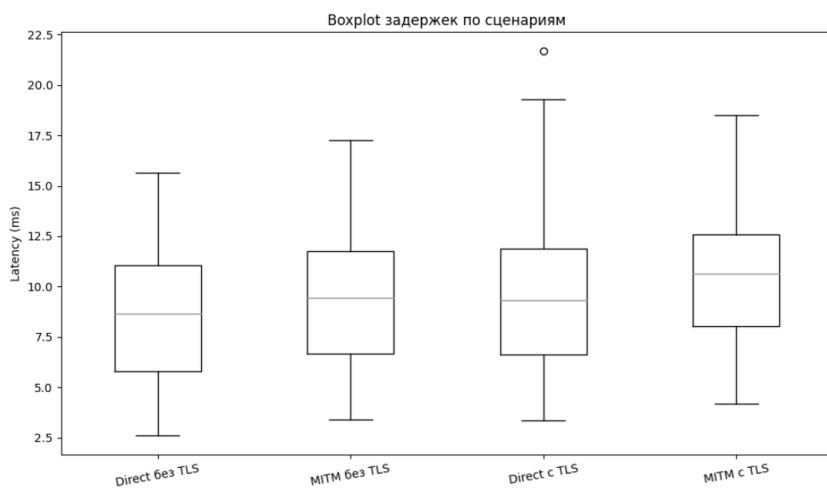


Рис. 11. Вохplot-диаграммы задержек по сценариям

Fig. 11. Boxplot-delay diagrams by scenario

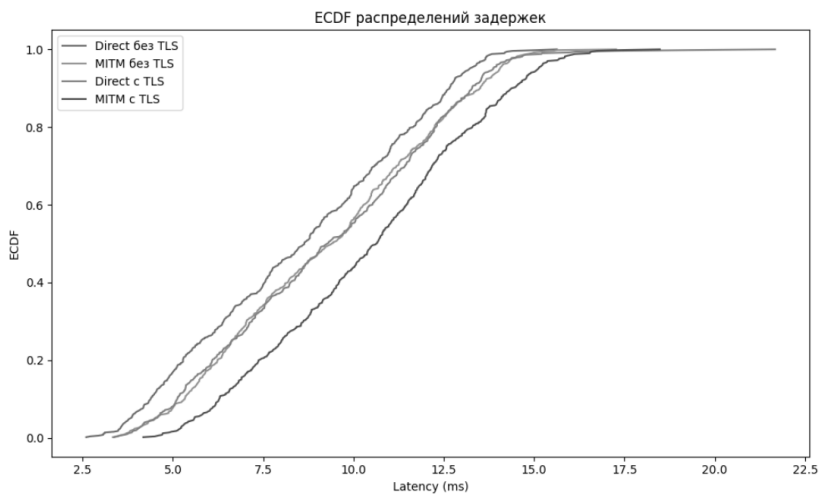


Рис. 12. Распределение задержек ECDF

Fig. 12. ECDF delay Allocation

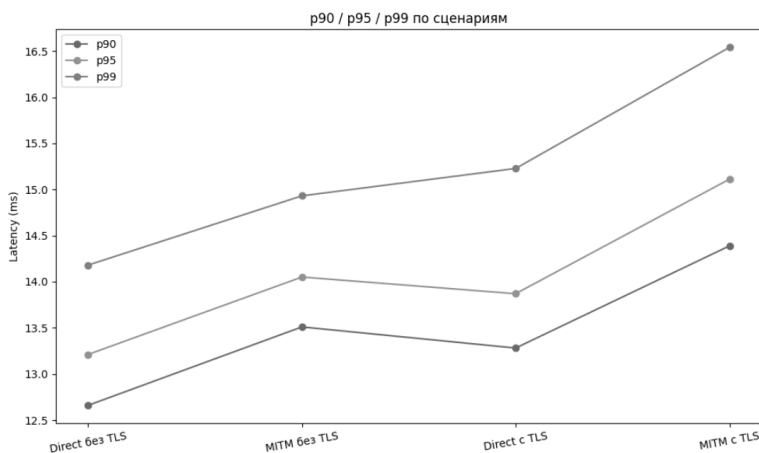


Рис. 13. Перцентили по сценариям

Fig. 13. Percentile scenarios

Данные соглашения показывают стабильную динамику показателей.

ЗАКЛЮЧЕНИЕ

В работе исследованы уязвимости каналов связи цифровых двойников в распределенных вычислительных системах на примере передачи телеметрии по протоколу MQTT. Экспериментально подтверждено, что незащищенный MQTT-канал (порт 1883) подвержен трем типам атак: MITM с подменой данных, Replay-атаке и DoS-атаке интенсивностью всего 100 сообщений в секунду, которая приводит к перегрузке брокера и рассинхронизации цифрового двойника. Предложен и апробирован комплекс практических мер защиты: шифрование MQTT over TLS (порт 8883) с проверкой сертификата сервера, что снижает вероятность чтения и модификации трафика при помощи MITM-прокси; модификация JavaScript-маппера Eclipse Ditto с отбрасыванием сообщений старше пяти секунд, значительно снижающая вероятность Replay-атаки; настройка правил iptables в цепочке DOCKER-USER (модули connlimit, hashlimit, recent), фильтрующая высокочастотный флуд без влияния на легитимный датчик.

Статистический анализ задержек MQTT publish RTT показал, что использование TLS увеличивает среднее время отклика на 9 %, а совместное применение TLS и промежуточного прокси – на 23 % относительно незащищенного канала, при этом форма распределения задержек сохраняется, что допустимо

для систем мягкого реального времени. Таким образом, комплексное применение TLS, проверки временных меток и ограничений на уровне межсетевого экрана обеспечивает надежную защиту каналов связи цифровых двойников от основных сетевых атак без нарушения их работоспособности.

СПИСОК ЛИТЕРАТУРЫ

1. A survey on digital twin for industrial internet of things: Applications, technologies and tools / H. Xu, J. Wu, Q. Pan, X. Guan, M. Guizani // *IEEE Communications Surveys & Tutorials*. – 2023. – Vol. 25 (4). – P. 2569–2598.
2. Attaran S., Attaran M., Celik B.G. Digital Twins and Industrial Internet of Things: Uncovering operational intelligence in industry 4.0 // *Decision Analytics Journal*. – 2024. – Vol. 10. – P. 100398. – DOI: 10.1016/j.dajour.2024.100398.
3. Soudan B. Cybersecurity of digital twins in industrial IoT environments // 2024 Advances in Science and Engineering Technology International Conferences (ASET). – IEEE, 2024. – P. 1–6. – DOI: 10.1109/ASET60340.2024.10708640.
4. MQTT vulnerabilities, attack vectors and solutions in the Internet of Things (IoT) / A.J. Hintaw, S. Manickam, M.F. Aboalmaaly, S. Karuppayah // *IETE Journal of Research*. – 2023. – Vol. 69 (6). – P. 3368–3397. – DOI: 10.1080/03772063.2021.1912651.
5. Dinculeană D., Cheng X. Vulnerabilities and limitations of MQTT protocol used between IoT devices // *Applied Sciences*. – 2019. – Vol. 9 (5). – P. 848. – DOI: 10.3390/app9050848.
6. Wong H.C. Man-in-the-Middle Attacks on MQTT based IoT networks. – Missouri University of Science and Technology, 2022.
7. Empl P., Pernul G. Digital-twin-based security analytics for the internet of things // *Information*. – 2023. – Vol. 14 (2). – P. 95. – DOI: 10.3390/info14020095.
8. Kampourakis K.E., Gkioulos V., Katsikas S. Cybersecurity Digital Twins for industrial systems: from literature synthesis to framework design // *Information*. – 2026. – Vol. 17 (3). – P. 286. – DOI: 10.3390/info17030286.
9. Методика оценки угроз безопасности информации: утв. ФСТЭК России 05.02.2021. – М., 2021.
10. Банк данных угроз безопасности информации ФСТЭК России. – URL: <https://bdu.fstec.ru/threat> (дата обращения: 05.20.2026).

Сильченко Никита Сергеевич, лаборант кафедры защиты информации Новосибирского государственного технического университета. Область научных интересов – безопасность каналов связи цифровых двойников, противодействие сетевым атакам. E-mail: silchenko.2022@stud.nstu.ru

Пермяков Руслан Анатольевич, старший преподаватель кафедры защиты информации Новосибирского государственного технического университета. Основное направление научных исследований – обеспечение информационной безопасности распределенных вычислительных систем и цифровых двойников. E-mail: permyakov@corp.nstu.ru

DOI: 10.17212/2782-2230-2026-3-24-40

Protection of digital twin communication channels from attacks in distributed computing systems*

N.S. Silchenko¹, R.A. Permyakov²

¹ *Novosibirsk State Technical University, 136 Nemirovich-Danchenko Street, Novosibirsk, 630087, Russian Federation, Laboratory assistant at the Department of Information Security. E-mail: silchenko.2022@stud.nstu.ru*

² *Novosibirsk State Technical University, 136 Nemirovich-Danchenko Street, Novosibirsk, 630087, Russian Federation, Senior Lecturer at the Department of Information Security. E-mail: permyakov@corp.nstu.ru*

Digital twin technology is being actively introduced into the industrial Internet of Things to monitor and manage complex cyber-physical systems in real time. The two-way nature of data exchange between a physical object and its virtual copy generates additional attack vectors, with attacks on communication channels using the lightweight MQTT protocol being particularly dangerous. The paper develops a model of information security threats for digital twin communication channels in accordance with the FSTEC methodology of Russia, and identifies six relevant threats, including data substitution, replay, and denial of service. An experimental stand has been created based on a temperature sensor emulator, the HiveMQ MQTT broker, and the Eclipse Ditto digital twin platform deployed in Docker containers. MITM attack scenarios with data substitution, Replay attacks and DoS attacks have been implemented. It is shown that an unsecured MQTT channel is vulnerable to all listed threats. A set of protection mechanisms is proposed: TLS channel encryption, timestamp verification in a JavaScript mapper, and iptables rules to limit traffic intensity. The effectiveness of the proposed measures has been experimentally confirmed: a MITM attack becomes impossible when the certificate is checked correctly, a Replay attack is prevented for the scenario under consideration by discarding outdated messages, and a DoS attack is significantly reduced due to filtering at the firewall level.

Keywords: digital twin, MQTT, TLS, MITM, Replay attack, DoS attack, iptables, Eclipse Ditto, HiveMQ, distributed computing systems, threat model

* Received 02 July 2026.

REFERENCES

1. Xu H., Wu J., Pan Q., Guan X., Guizani M. A survey on digital twin for industrial internet of things: Applications, technologies and tools. *IEEE Communications Surveys & Tutorials*, 2023, vol. 25 (4), pp. 2569–2598.
2. Attaran S., Attaran M., Celik B.G. Digital Twins and Industrial Internet of Things: Uncovering operational intelligence in industry 4.0. *Decision Analytics Journal*, 2024, vol. 10, p. 100398. DOI: 10.1016/j.dajour.2024.100398.
3. Soudan B. Cybersecurity of digital twins in industrial IoT environments. *2024 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE, 2024, pp. 1–6. DOI: 10.1109/ASET60340.2024.10708640.
4. Hintaw A.J., Manickam S., Aboalmaaly M.F., Karuppayah S. MQTT vulnerabilities, attack vectors and solutions in the Internet of Things (IoT). *IETE Journal of Research*, 2023, vol. 69 (6), pp. 3368–3397. DOI: 10.1080/03772063.2021.1912651.
5. Dinculeană D., Cheng X. Vulnerabilities and limitations of MQTT protocol used between IoT devices. *Applied Sciences*, 2019, vol. 9 (5), p. 848. DOI: 10.3390/app9050848.
6. Wong H.C. *Man-in-the-Middle Attacks on MQTT based IoT networks*. Missouri University of Science and Technology, 2022.
7. Empl P., Pernul G. Digital-twin-based security analytics for the internet of things. *Information*, 2023, vol. 14 (2), p. 95. DOI: 10.3390/info14020095.
8. Kampourakis K.E., Gkioulos V., Katsikas S. Cybersecurity Digital Twins for industrial systems: From literature synthesis to framework design. *Information*, 2026, vol. 17 (3), p. 286. DOI: 10.3390/info17030286.
9. FSTEC of Russia. *Metodika otsenki ugroz bezopasnosti informatsii* [Methodology for assessing information security threats]. Moscow, 2021.
10. FSTEC of Russia. *Information security threats database*. (In Russian). Available at: <https://bdu.fstec.ru/threat> (accessed 20.05.2026).

Для цитирования:

Сильченко Н.С., Пермяков Р.А. Защита каналов связи цифровых двойников от атак в распределенных вычислительных системах // Безопасность цифровых технологий. – 2026. – № 3 (122). – С. 24–40. – DOI: 10.17212/2782-2230-2026-3-24-40.

For citation:

Silchenko N.S., Permyakov R.A. Zashchita kanalov svyazi tsifrovyykh dvoynikov ot atak v raspredelennykh vychislitel'nykh sistemakh [Protection of digital twin communication channels from attacks in distributed computing systems]. *Bezopasnost' tsifrovyykh tekhnologii = Digital Technology Security*, 2026, no. 3 (122), pp. 24–40. DOI: 10.17212/2782-2230-2026-3-24-40.