nettle в НАЙС.ОС - полная поддержка ГОСТ

Введение

Алгоритмы **Кузнечик** и **Магма**, входящие в состав стандарта **ГОСТ Р 34.12-2015**, представляют собой современные российские симметричные блочные шифры. Они пришли на смену устаревшему ГОСТ 28147-89 и стали обязательными для применения в системах, требующих соответствия требованиям **ФСТЭК** и **ГОСТ Р ИСО/МЭК 15408**.

Шифр **Магма** (64-битный блок) является переработанной версией ГОСТ 28147-89 с фиксированным S-box и улучшенной структурой. **Кузнечик** — это более современный и надёжный 128-битный блочный шифр, разработанный специально для повышения уровня криптографической стойкости и эффективности реализации как в программных, так и в аппаратных системах.

Зачем они нужны в Linux-дистрибутиве?

- ✅ **Для совместимости с российскими криптосервисами** — большинство систем ЭДО, VPN, защищённой связи и подписи требуют использование ГОСТ-алгоритмов.
- ✅ **Для построения защищённых каналов и хранилищ** — особенно в системах, подлежащих сертификации или использующихся в рамках импортозамещения.
- ✅ **Для построения полноценного ГОСТ-криптостека** — в связке с ГОСТ-хэшами, подписью и VKO, Кузнечик и Магма позволяют закрыть полный контур криптографической безопасности.

Ключевое событие: начиная с этой версии, в **НАЙС.ОС** добавлена полная штатная поддержка алгоритмов **Кузнечик** и **Магма** в библиотеке `libnettle`, что позволяет использовать их в TLS, подписи, шифровании и других прикладных сценариях **из коробки** — без установки стороннего ПО.

Это означает, что ГОСТ-алгоритмы из **ГОСТ Р 34.12-2015** теперь полностью доступны для всех систем и приложений, использующих `libgnutls`, `libnettle`, `curl`, `lighttpd`, `exim` и

другие криптографические или TLS-инструменты.

✍ В следующих разделах мы рассмотрим, как была достигнута эта интеграция, какие алгоритмы уже были в апстриме, какие добавлены разработчиками НАЙС.ОС, и что это даёт конечным пользователям на практике.

Что было в апстриме (Nettle и GnuTLS)

Nettle — это легковесная, модульная криптографическая библиотека, широко используемая в проектах с открытым исходным кодом. Она реализует набор базовых алгоритмов (хэши, шифры, подписи и пр.), обеспечивая при этом **низкоуровневую производительность** и **гибкость для интеграции**.

Одним из ключевых потребителей Nettle является **GnuTLS** — TLS-библиотека, используемая в таких инструментах, как `curl`, `wget`, `exim`, `lighttpd` и других. Именно GnuTLS отвечает за реализацию TLS/SSL-протоколов в **НАЙС.ОС** и тесно зависит от функций, предоставляемых Nettle.

🔍 Поддержка ГОСТ в апстриме Nettle

В официальной (апстримной) версии библиотеки Nettle уже присутствовала частичная поддержка российских криптографических алгоритмов, в том числе:

- ✅ **ГОСТ 28147-89** — классический 64-битный блочный шифр, используемый в ряде наследуемых систем.
- ✅ **ГОСТ Р 34.11-94** — устаревшая хэш-функция, ранее использовавшаяся с ГОСТ 28147-89.
- ✅ **Стрибог (ГОСТ Р 34.11-2012)** — современная хэш-функция, реализованная в вариантах 256 и 512 бит.
- ✅ **ГОСТ Р 34.10-2012** — алгоритм цифровой подписи на эллиптических кривых.
- ✅ **VKO (ГОСТ R 34.10-2012)** — алгоритм выработки общего ключа на основе ЭЦП и хэша.

✍ Эти алгоритмы присутствовали в библиотеке Nettle и могли использоваться приложениями, работающими с открытым ГОСТ-криптостеком.

⊘ Чего не было в апстриме

Несмотря на частичную поддержку ГОСТ, в кодовой базе Nettle отсутствовали **два ключевых алгоритма из ГОСТ Р 34.12-2015**, необходимых для реализации

современных криптографических протоколов:

- **× Кузнечик (GOST R 34.12-2015)** — 128-битный блочный шифр, утверждённый как замена ГОСТ 28147-89 и обязательный для новых систем.
- **× Магма (GOST R 34.12-2015)** — переработанный 64-битный шифр, совместимый с ГОСТ 28147-89, но с фиксированным S-box и улучшенными параметрами.

⚠ Хотя в **GnuTLS** предусматривалась поддержка Кузнечика и Магмы, она **не была активна** по умолчанию, так как напрямую зависит от наличия реализаций в библиотеке Nettle.

В частности, в заголовочном файле `gnutls.h.in` уже были описаны следующие алгоритмы и режимы:

```
// Фрагменты из GnuTLS
@GNUTLS_CIPHER_KUZNYECHIK_CTR_ACPKM     // Кузнечик в режиме CTR-ACPKM
@GNUTLS_CIPHER_MAGMA_CTR_ACPKM         // Магма в режиме CTR-ACPKM
@GNUTLS_MAC_KUZNYECHIK_OMAC            // MAC по Кузнечику
@GNUTLS_MAC_MAGMA_OMAC                 // MAC по Магме
```

Однако без поддержки этих шифров в Nettle, GnuTLS не мог их использовать в полной мере. Таким образом, возможности TLS-шифрования по ГОСТ Р 34.12-2015 были заблокированы.

Чтобы активировать полную поддержку Кузнечика и Магмы, необходимо было реализовать их внутри `libnettle` и зарегистрировать как поддерживаемые `cipher`-объекты.

Что было сделано в НАЙС.ОС

В рамках развития отечественной криптографической поддержки в **НАЙС.ОС** разработчики системы выполнили **полную и чистую интеграцию алгоритмов ГОСТ Р 34.12-2015** — **Кузнечика** и **Магмы** — в криптографическую библиотеку `libnettle`.

В отличие от апстрима, где эти алгоритмы отсутствовали, в **НАЙС.ОС** реализация была произведена вручную, **с нуля**, с соблюдением всех архитектурных и инженерных стандартов проекта Nettle:

- ✓ **Реализация алгоритма Кузнечик** (128-битный блочный шифр) с полной поддержкой ключевых операций, форматами LE/BE, внутренним состоянием и режимами блочного шифрования.

- ✅ **Реализация алгоритма Магма** (64-битный блочный шифр) на основе современного представления ГОСТ 28147-89 с фиксированным S-box TC26-Z.
- ✅ **Тестирование** на основе официальных ГОСТ-векторов (в том числе из СП 800.38A и RFC 7836) с использованием внутренней тест-системы `testsuite/` библиотеки Nettle.
- ✅ **Интеграция с мета-структурами** (`nettle-meta.h`, `nettle-meta-ciphers.c`) — добавлены объекты `nettle_kuznyechik` и `nettle_magma`.
- ✅ **Добавление в `nettle-benchmark`** — теперь алгоритмы Кузнечик и Магма участвуют в сравнительных тестах производительности.
- ✅ **Активирована поддержка в GnuTLS** — через механизмы auto-detection и условной компиляции (`ENABLE_GOST`), новые алгоритмы автоматически доступны для TLS 1.2.

Реализация успешно прошла все внутренние тесты, включая:

- блочное шифрование и расшифровку с проверкой выходных векторов;
- регрессионные тесты;
- интеграционные тесты с GnuTLS;
- проверку на совместимость с CryptoPro/TC26-спецификацией.

Благодаря аккуратной и строго соответствующей upstream-архитектуре реализации, алгоритмы полностью интегрированы в крипто-экосистему **НАЙС.ОС** и доступны **из коробки** — без необходимости в дополнительной установке, сборке или модификации сторонних библиотек.

🎉 С этого момента любые приложения, использующие GnuTLS или напрямую работающие с `libnettle` (например, `curl`, `wget`, `lighttpd`, `exim`), получают доступ к ГОСТ Р 34.12-2015 **автоматически**.

Что теперь входит в НАЙС.ОС

После интеграции симметричных шифров **Кузнечик** и **Магма**, в **НАЙС.ОС** теперь доступен **полный стек ГОСТ-алгоритмов** прямо "из коробки" — без необходимости доустановки или ручной настройки. Всё включено на уровне системных библиотек `libnettle` и `libgnutls`.

Благодаря этому любой пользователь или разработчик может использовать ГОСТ-шифры, хэши, подписи и механизмы TLS сразу после установки системы — как в пользовательских приложениях, так и на уровне инфраструктурных компонентов:

VPN, TLS-серверов, системных служб и контейнерных окружений.

🎁 Поддержка ГОСТ в НАЙС.ОС

Категория	Алгоритм	ГОСТ / стандарт	Поддержка в НАЙС.ОС
🔐 Шифры	`ГОСТ 28147-89`	ГОСТ Р 34.12-89	✅ Да
	`Магма`	ГОСТ Р 34.12-2015	✅ Да
	`Кузнечик`	ГОСТ Р 34.12-2015	✅ Да
Хэши	`ГОСТ R 34.11-94`	ГОСТ Р 34.11-94	✅ Да
	`Стрибог 256/512`	ГОСТ Р 34.11-2012	✅ Да
🛡️ MAC	`HMAC ГОСТ-94, Стрибог, IMIT`	ГОСТ Р 34.11 / 34.12	✅ Да
✍️ Подписи	`ГОСТ 34.10-2001 / 2012` (256/512)	ГОСТ Р 34.10-2012	✅ Да
🔁 Обмен ключами	`VKO`	ГОСТ Р 34.10-2012	✅ Да
🌐 TLS	`GOST TLS 1.2` через `GnuTLS`	ГОСТ-совместимые ciphersuites	✅ Да

✅ Все перечисленные алгоритмы доступны сразу после установки **НАЙС.ОС**, без ручной компиляции, дополнительных библиотек или сторонних модулей.

🔍 Проверка доступности

После установки системы вы можете убедиться в наличии алгоритмов с помощью команд:

```
gnutls-cli --list --priority GOST
grep -i gost /proc/crypto
openssl list -cipher-algorithms | grep gost
```

💡 **Подсказка:** при использовании `gnutls-cli` и `curl` в НАЙС.ОС теперь можно указывать приоритеты TLS с поддержкой ГОСТ, например:
`curl --gnutls --ciphersuite 'NONE:+VERS-TLS1.2:+GOST'`

Что это даёт пользователям НАЙС.ОС

Благодаря полной встроенной поддержке ГОСТ-алгоритмов, пользователи **НАЙС.ОС** получают полноценную **криптографическую платформу отечественного уровня** без необходимости установки сторонних компонентов. Всё уже встроено, протестировано и готово к использованию в любых задачах — от защищённых соединений до документооборота.

✔ Полная ГОСТ-криптография без внешних зависимостей

- **Никаких OpenSSL-плагинов**, проприетарных библиотек или CryptoPro.
- 🔗 **Всё работает через** `GnuTLS` и `Nettle`, уже включённые в систему.
- 🔍 Проверка и использование возможны в любых приложениях, поддерживающих GnuTLS (в том числе curl, wget, git).

✔ ГОСТ-TLS для защищённых соединений

- 🔐 **Поддержка ГОСТ ciphersuites в TLS 1.2**, включая шифры на базе Магмы и Кузнечика, хэши Стрибог, подписи ГОСТ.
- ⚙️ **Использование в приложениях:** `curl`, `lighttpd`, `openconnect`, `exim`, `git-remote-http`, `libvirt` и др.
- 💡 Настройка приоритетов через GnuTLS-профиль: `'NONE:+VERS-TLS1.2:+GOST'`

✔ ГОСТ-подписи и VKO

- ✍️ **Подписание файлов и сообщений** через ГОСТ R 34.10-2012 (256/512 бит).
- 🔁 **Обмен ключами** через VKO ГОСТ 2012 с использованием отечественных эллиптических кривых.
- Поддержка форматов CMS, PKCS#7, X.509 с ГОСТ-криптографией.

✔ Электронный документооборот и ГОСТ-PKI

- 📄 **Хэширование и подписание PDF, XML, JSON** по требованиям ФСТЭК и Минцифры.
- 🏛️ **Создание ГОСТ-совместимой PKI** для внутреннего документооборота, электронных доверенностей и ЭП.
- ⚖️ Соответствие требованиям к СКЗИ, в том числе без использования проприетарного ПО.

✔ Импортозамещение и соответствие требованиям

- **Полное импортозамещение:** все алгоритмы реализованы на **открытом коде**, без "чёрных ящиков".
- 🔐 ГОСТ реализован в системных библиотеках `libnettle` и `libgnutls`, прошёл тестирование и доступен в пользовательских приложениях.
- ✔ **Готовность к сертификации** по требованиям ФСТЭК/ФСБ (при наличии режима контроля целостности и ограничений по TLS).

🛰 Благодаря этой интеграции, **НАЙС.ОС** стал одной из немногих российских ОС, в которой ГОСТ-криптография реализована полностью, открыто и доступна всем пользователям без дополнительных лицензий.

Как это работает из коробки

В **НАЙС.ОС** поддержка ГОСТ реализована **на уровне системных библиотек**, без внешних плагинов, закрытых компонентов или ручной настройки. Всё уже включено в базовую систему и доступно сразу после установки.

🎁 Установленные компоненты

- ✔ **libnettle** — содержит полную реализацию ГОСТ-алгоритмов: Магма, Кузнечик, ГОСТ 28147-89, Стрибог, ГОСТ DSA и VKO.
- ✔ **libgnutls** — собрана с опцией `--enable-gost`, что активирует ГОСТ-поддержку в TLS и криптографических утилитах.
- ✔ **certtool** — поддерживает генерацию ГОСТ-сертификатов, создание ключей, формирование PKCS#10-запросов.

🔐 Активация ГОСТ в TLS

ГОСТ-алгоритмы доступны в `gnutls-cli`, `curl`, `openconnect` и других клиентах, использующих GnuTLS. Для включения достаточно указать приоритет TLS-схем:

```
gnutls-cli --priority 'NONE:+VERS-TLS1.2:+GOST' example.gov.ru
```

Это включает только ГОСТ-совместимые алгоритмы в TLS 1.2 и гарантирует соответствие российским криптографическим стандартам.

i ГОСТ-алгоритмы в GnuTLS недоступны для TLS 1.3 — согласно документации, при использовании ГОСТ рекомендуется отключать TLS 1.3.

🛠 Работа с ГОСТ-сертификатами

Утилита `certtool` в НАЙС.ОС умеет:

- ✍ Создавать ключи ГОСТ R 34.10-2012 (256/512 бит)
- 📄 Генерировать запросы PKCS#10 с ГОСТ-подписями
- Подписывать и выдавать сертификаты с ГОСТ-параметрами

```
certtool --generate-privkey --key-type gost12-256 --outfile gost.key
certtool --generate-request --load-privkey gost.key --outfile gost.req
certtool --generate-certificate --load-ca-privkey ca.key --load-ca-certificate ca.pem \
        --load-request gost.req --outfile gost.crt
```

🧿 Ничего дополнительно делать не нужно

- **× Не нужно** устанавливать проприетарные модули или CryptoPro
- **× Не нужно** патчить ядро или libssl
- ✓ **Всё работает** через штатные инструменты и пакеты **из основного репозитория НАЙС.ОС**

Благодаря полной интеграции ГОСТ в **libnettle** и **GnuTLS**, система **НАЙС.ОС** готова для применения в защищённых информационных системах (ЗИС), в корпоративной и государственной инфраструктуре.

7. Примеры использования

После установки **НАЙС.ОС** все ГОСТ-алгоритмы доступны без дополнительной настройки. Ниже приведены практические примеры использования встроенных криптографических инструментов:

📡 Тестирование ГОСТ-TLS через `gnutls-cli` и `gnutls-serv`

Сначала сгенерируем ГОСТ-ключ и самоподписанный сертификат:

```
certtool --generate-privkey --key-type gost12-256 --outfile gost.key

certtool --generate-self-signed \
  --load-privkey gost.key \
  --template <<EOF
cn = "localhost"
expiration_days = 365
tls_www_server
dns_name = "localhost"
```

```
EOF
 --outfile gost.crt
```

Теперь запускаем тестовый TLS-сервер:

```
gnutls-serv --priority "NONE:+VERS-TLS1.2:+GOST" \
 --x509certfile gost.crt --x509keyfile gost.key --port 5555
```

И подключаемся к нему:

```
gnutls-cli --priority "NONE:+VERS-TLS1.2:+GOST" -p 5555 localhost
```

Если всё корректно — будет установлен TLS 1.2-соединение с использованием ГОСТ-шифров.

✍ **Подпись и проверка файла по ГОСТ Р 34.10-2012**

Создадим ГОСТ-ключ и подпишем файл:

```
certtool --generate-privkey --key-type gost12-256 --outfile gost.key

openssl dgst -sign gost.key -binary -sha256 -out signature.bin somefile.txt
```

Проверим подпись (требуется открытый ключ):

```
openssl pkey -in gost.key -pubout -out pubkey.pem

openssl dgst -verify pubkey.pem -signature signature.bin -sha256 somefile.txt
```

i В НАЙС.ОС используется GnuTLS и Nettle, однако openssl с ГОСТ-патчами также может быть доступен в отдельном репозитории.

🔒 **Шифрование и расшифровка с помощью Кузнечика**

Для простого примера воспользуемся `nettle`-утилитой `nettle-benchmark`, а для ручного шифрования используем OpenSSL (при наличии ГОСТ-поддержки) или напишем скрипт с использованием `gost-crypto` через Python.

Пример ручного шифрования файла с помощью OpenSSL (если собран с ГОСТ):

```
openssl enc -k secretpassword \
```

```
 -in secret.txt -out secret.enc \
 -cipher grasshopper -p -nosalt
```

И расшифровка:

```
openssl enc -d -k secretpassword \
 -in secret.enc -out decrypted.txt \
 -cipher grasshopper -nosalt
```

⚠ Для полного ГОСТ-шифрования через openssl требуется его сборка с плагином `engine-gost`. В НАЙС.ОС рекомендуем использовать стандартные инструменты с **GnuTLS/Nettle**.

🏢 Тестирование производительности

Для оценки производительности алгоритмов используйте встроенный `nettle-benchmark`:

```
nettle-benchmark | grep -E 'gost|magma|kuznyechik'
```

Это покажет скорость шифрования ГОСТ-алгоритмами на вашей системе.

✓ Все алгоритмы ГОСТ работают в **user-space**, не требуют модулей ядра и полностью соответствуют **ФСТЭК**-требованиям.

8. Заключение

НАЙС.ОС стала одной из первых отечественных операционных систем, в которой **реализован полный стек ГОСТ-криптографии**, включая новейшие алгоритмы **Кузнечик** и **Магма** по ГОСТ Р 34.12-2015 — **прямо из коробки**, без установки дополнительных компонентов.

Все алгоритмы реализованы **встроенно** в системные библиотеки `libnettle` и `GnuTLS`, без внешних зависимостей, проприетарных плагинов или стороннего кода.

🎁 Полный стек ГОСТ

- ✓ Блочные шифры: ГОСТ 28147-89, Магма, Кузнечик
- ✓ Хэши: ГОСТ R 34.11-94, Стрибог 256/512
- ✓ MAC: IMIT, HMAC, CMAC на базе ГОСТ
- ✓ Подписи: ГОСТ R 34.10-2001 / 2012 (256/512)

- ✅ Обмен ключами: VKO на ГОСТ-кривых
- ✅ TLS: ГОСТ-наборы шифров через GnuTLS

🏛️ Для кого это важно?

- 🛡️ **Регуляторам и ведомствам** — соответствие требованиям ФСТЭК, ГОСТ, ТК26.
- 💼 **Корпоративным системам** — возможность безопасной электронной подписи, TLS, обмена ключами и защиты хранимых данных.
- **СКЗИ и встроенным системам** — компактные реализации ГОСТ без лишних зависимостей.
- 🔍 **Исследователям и разработчикам** — открытая архитектура, легко проверяемая и расширяемая.

🚀 Преимущества подхода НАЙС.ОС

- 🔓 **Открытый код** — всё реализовано в свободных библиотеках Nettle и GnuTLS.
- ⚙️🚫 **Никаких бинарных blob'ов**, проприетарных плагинов или лицензий.
- **Интеграция на уровне системы** — всё работает с утилитами из репозитория.
- 🛠️🚫 **Готово для применения** — ГОСТ TLS, подписи, хранилища и проверка целостности.

✅ ГОСТ-криптография в НАЙС.ОС — это **готовая платформа** для построения сертифицируемых решений, доверенных систем и безопасной ИТ-инфраструктуры — на базе современных открытых технологий.