Разработка проектов рекомендаций по стандартизации криптографических механизмов для реализации сервисов электронной коммерции в значимых платежных системах Российской Федерации

Специалист по защите информации — **Елена Николаевна Шкоркина**
shkorkina@systempb.ru

Руководитель отдела системных исследований — Алла Геннадьевна Герасимова

Руководитель направления перспективных проектов — Александр Александрович Габов

Требования к криптографическим алгоритмам СКЗИ платежных систем РФ

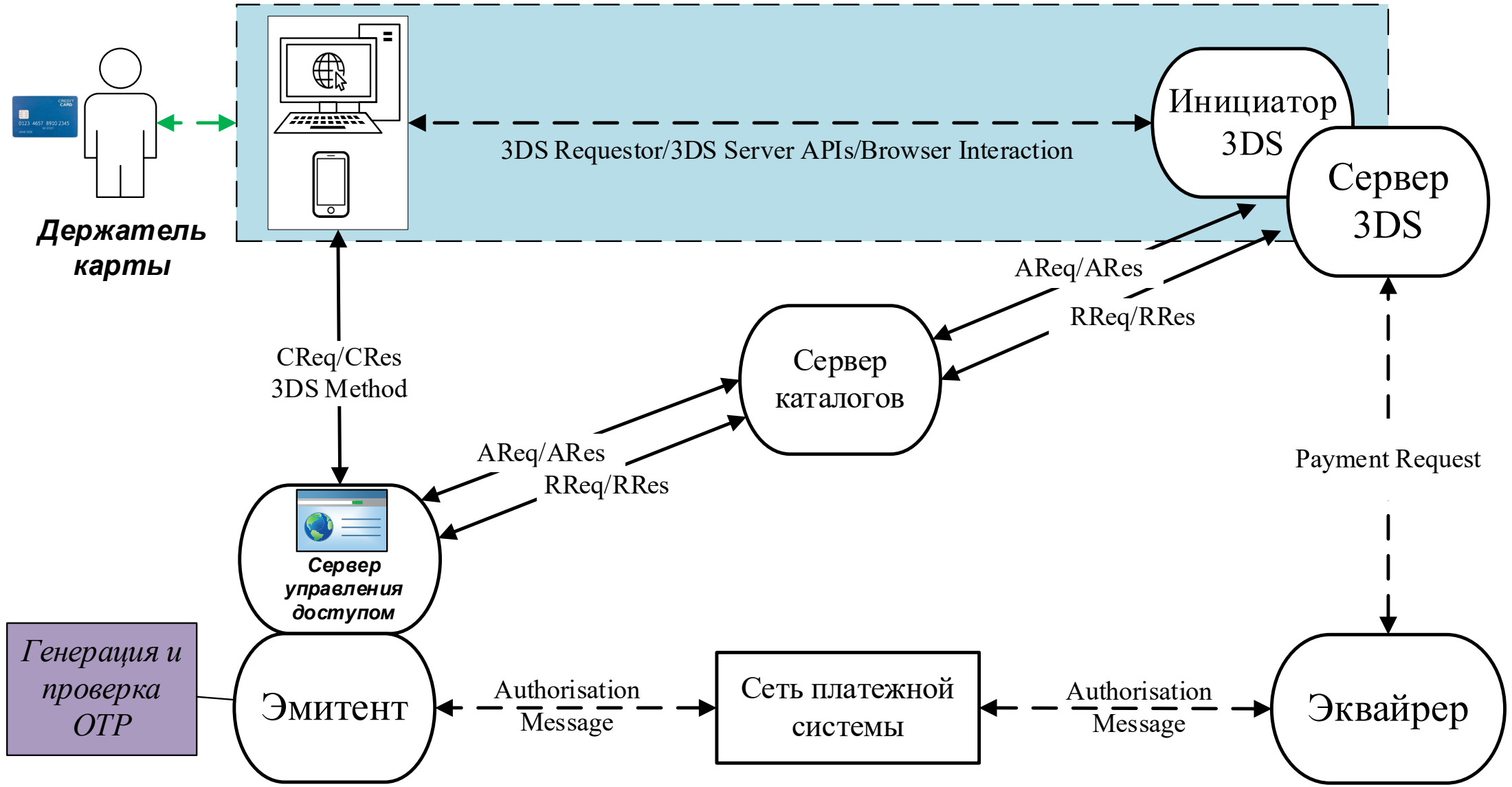
Положение Банка России № 719-П (от 04.06.2020): СКЗИ из состава аппаратных модулей безопасности, реализующие **иностранные криптографические алгоритмы и криптографические алгоритмы РФ, должны** иметь подтверждение соответствия требованиям, установленным ФОИВ в области обеспечения безопасности

Требования ФОИВ к криптомеханизмам СКЗИ:
1) Должны использоваться криптографические механизмы из числа национальных стандартов РФ или рекомендаций по стандартизации, или криптографические механизмы, имеющие положительное заключение по результатам их экспертных криптографических исследований.
2) С целью обеспечения совместимости должны использоваться криптографические механизмы, отвечающие международным стандартам (ISO).

Требуется разработка национальных криптографических механизмов, в том числе:

- формирования одноразовых паролей (OTP, one-time password);
- управления ключами транзакций при оказании услуг электронной коммерции (Derived unique key per transaction, DUKPT);
- ключевого контейнера ACS X9 TR 31.

OTP: использование в рамках протокола 3-D Secure 2.0



OTP: функция генерации

Функция генерации одноразового пароля имеет вид:

$$otp = OTP(K_{uid}, InputData, n) = DStr^n\left(Int\left(Func(K_{uid}, InputData)\right) mod\ 10^n\right),$$

в котором

$Int()$ – функция, ставящая в соответствие битовой строке число;

$DStr^n()$ – функция, ставящая в соответствие числу десятичную строку;

K_{uid} – ключ псевдослучайной функции (256 бит);

$InputData$ – входные данные произвольной длины (`counter_value`, `time_value`, `pwd_value`, `transaction_info`);

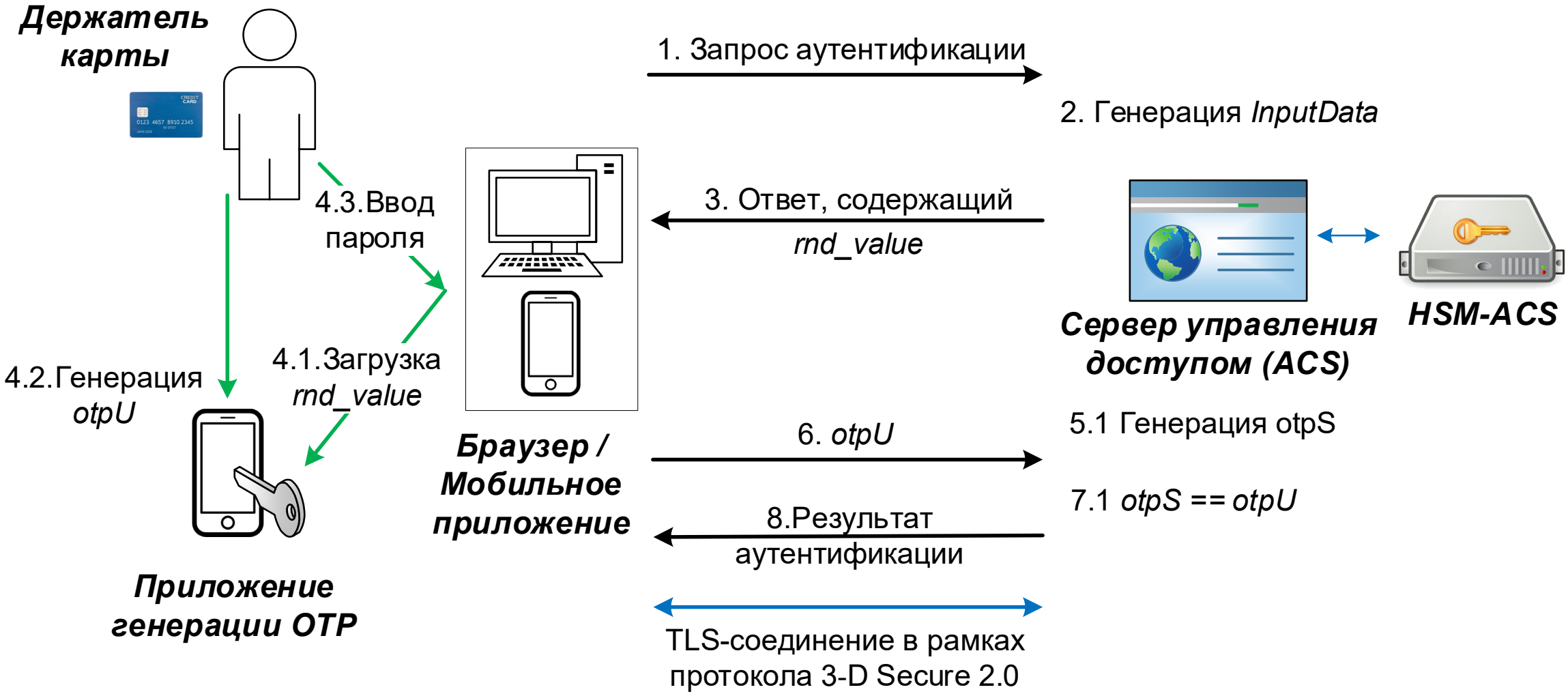
n – длина одноразового пароля otp;

$Func(K_{uid}, InputData)$ – ключевая псевдослучайная функция:

1) алгоритм HMAC_GOSTR3411_2012_256, описанный в Р 50.1.113–2016;
2) алгоритм выработки имитовставки, определенный в ГОСТ Р 34.13-2018, с длиной имитовставки, равной длине блока используемого блочного шифра ($64/128$ бит).

СПБ

OTP: схема аутентификации



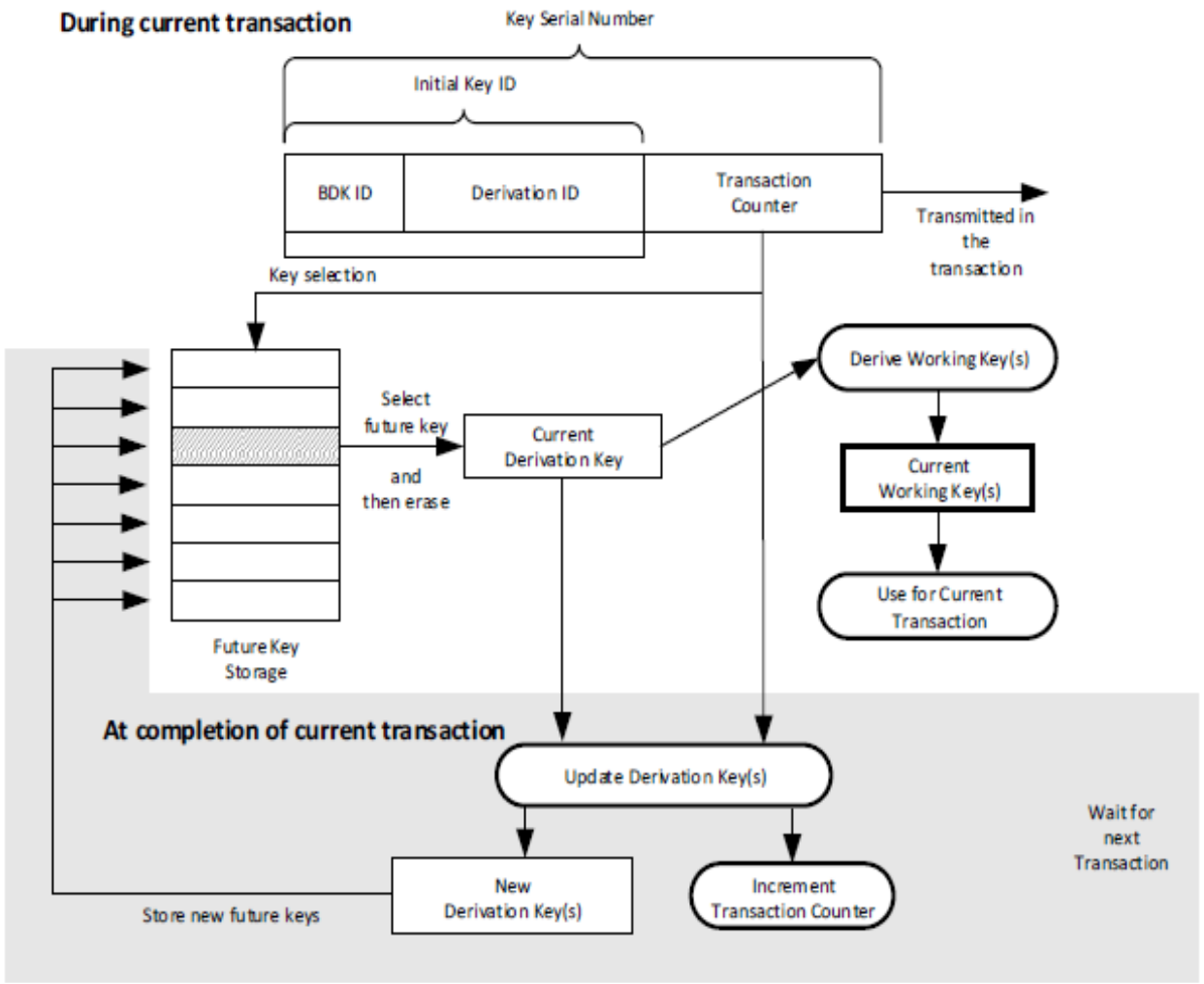
DUKPT: требования и иерархия ключей

DUKPT (ANSI X9.24-3-2017) – механизм получения ключа (ключей) из начального терминального ключа DUKPT на основе номера транзакции таким образом, чтобы:

- терминал не сохранял никакой информации, которая могла бы быть использована для получения ключа транзакции после завершения транзакции
- модуль безопасности (HSM) на приемной стороне мог бы получить тот же самый ключ транзакции с учетом требований по быстродействию.

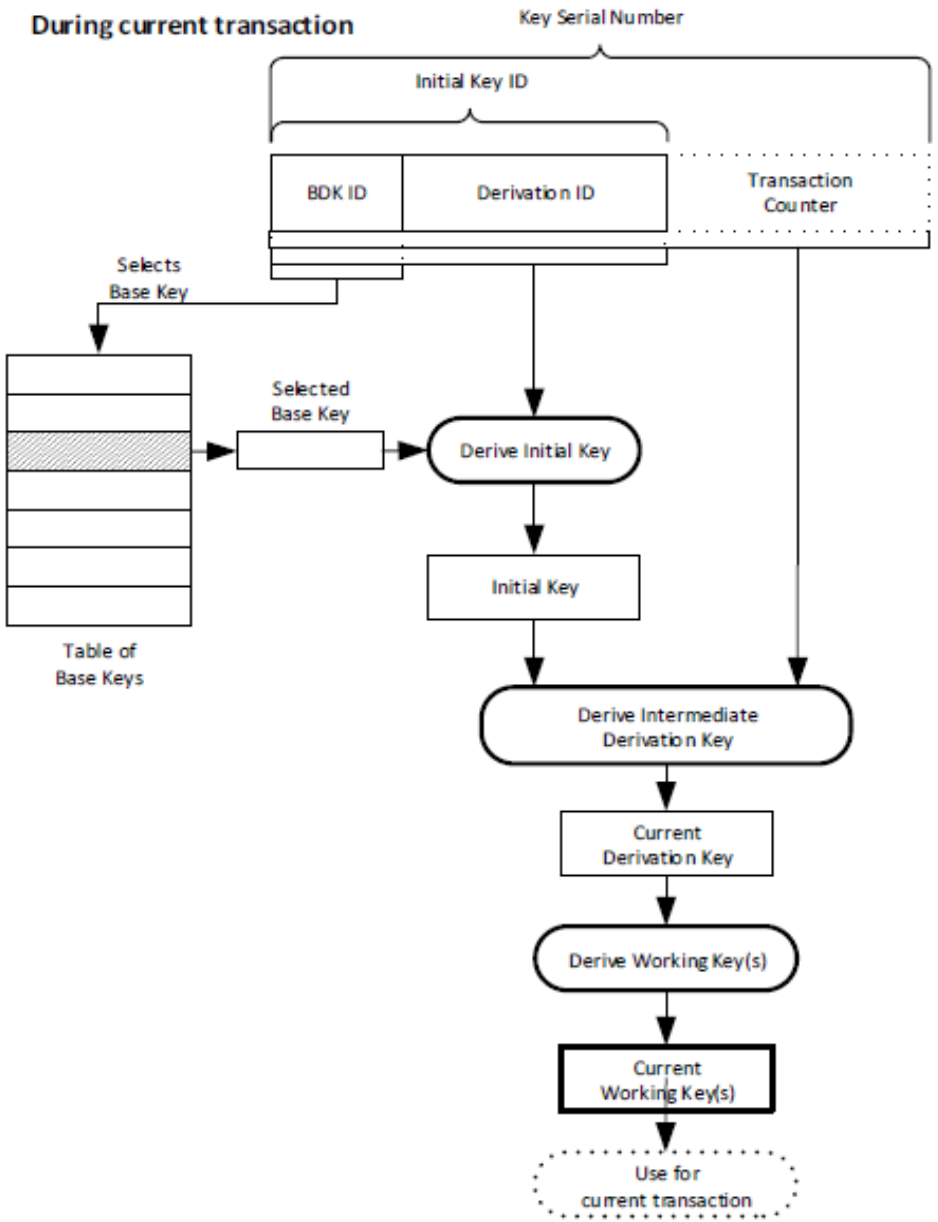
Обозначение	Название	Назначение	Идентификатор
BDK - Base Derivation Key	Базовый ключ диверсификации	Выработка уникального исходного ключа диверсификации СКЗИ, инициирующего транзакцию	*BDK ID*
IK - Initial Key	Исходный ключ диверсификации	Выработка текущего ключа транзакции	*Initial Key ID = BDK ID || Derivation ID*
CDK - Current Derivation Key	Текущий ключ диверсификации	Выработки текущих рабочих ключей	*Key Serial Number (KSN)= Initial Key ID || Transaction Counter*
CWK - Current Working Key(s)	Рабочие ключи транзакции	Защита PIN, шифрование информации или подсчет MAC	*Идентифицируются по KSN и Key Usage*

Выработка ключей в СКЗИ, инициирующем транзакцию



- По текущему значению счетчика транзакции (*TC*) выбирается соответствующий текущий ключ транзакции (*CDK*);
- С использованием выбранного ключа *CDK* формируются необходимые текущие рабочие ключи (*CWK*);
- Обработка транзакции с использованием текущих рабочих ключей в соответствии с их назначением;
- Удаление выбранного текущего ключа *CDK* (а также все неиспользованные ключи *CDK*, соответствующие меньшим значениям *Transaction Counter*, если таковые имеются) из хранилища будущих ключей *CDK;*
- Выполняется «сдвиг» будущих ключей *CDK* в хранилище, а также формируются и помещаются в хранилище новые ключи *CDK* для освободившихся позиций.
- Инкрементируется счетчик транзакций *TC.*

Выработка ключей в СКЗИ, обрабатывающем транзакцию

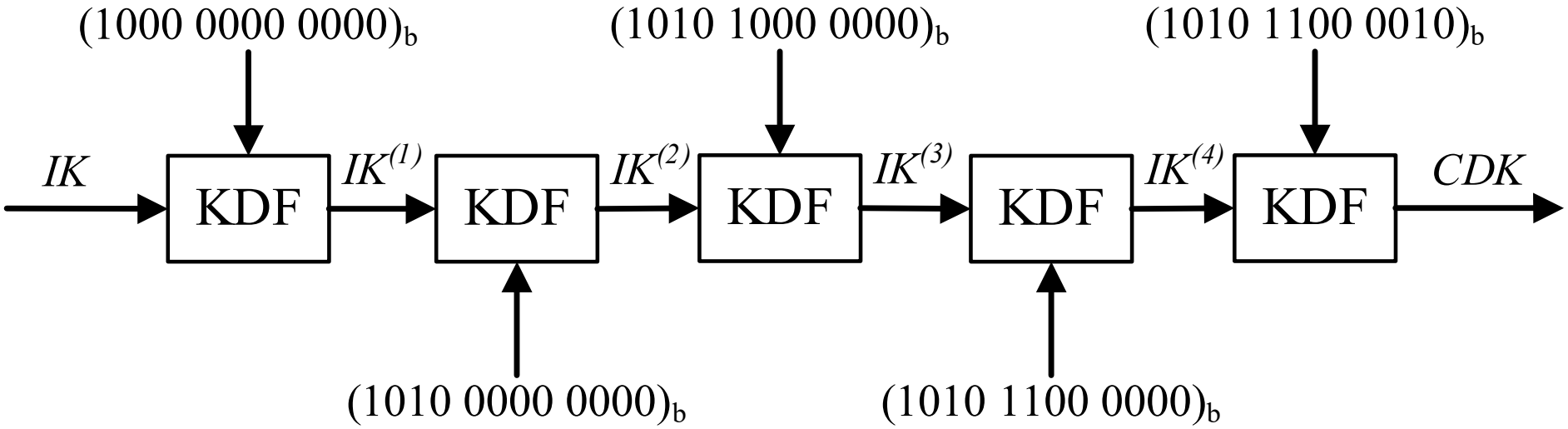


- По серийному номеру ключа (*KSN*) определяется базовый ключ диверсификации (*BDK*);
- С использованием *BDK* формируется исходный ключ диверсификации (*IK*);
- С использованием значения счетчика транзакции (*TC*) формируется текущий ключ диверсификации (*CDK*);
- С использованием сформированного ключа *CDK* формируются необходимые текущие рабочие ключи (*CWK*);
- Используются текущие рабочие ключи для обработки транзакции в соответствии с их назначением.

Алгоритм формирования текущего ключа диверсификации

Для получения ключа CDK HSM должен выполнить столько операций диверсификации, сколько битов "1" получено в значении счетчика транзакций.

Пример работы алгоритма в случае получения счетчика транзакции, равного $(1010\ 1100\ 0010)_b$ (используется счетчик короче фактического значения):



Текущий ключ транзакции (CDK) используется для выработки рабочих ключей (CWK).

Функции диверсификации ключей DUKPT

AES DUKPT (ANSI X9.24-3-2017)	GOST DUKPT (проект)
AES DUKPT использует NIST 800-108 KDF (NIST SP 800-108: Recommendation for Key Derivation using Pseudorandom Functions) в режиме счетчика с AES-ECB в качестве базовой функции для получения ключей, эквивалентных одному выходному блоку, и AES-CMAC для получения ключей, кратных размеру выходного блока	Функция вычисления имитовставки (ГОСТ Р 34.13-2015) на основе блочного шифра «Кузнечик» (ГОСТ Р 34.12-2015)

Ключевой контейнер ACS X9 TR 31 – 2018

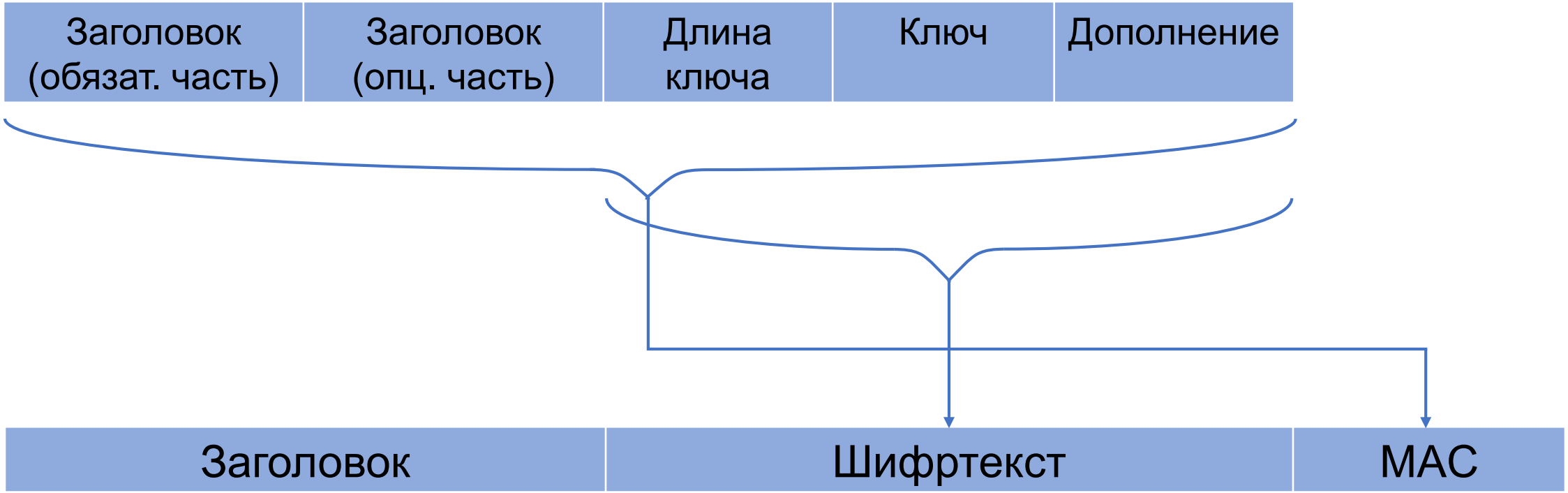
Разрабатывался как дополнение к стандарту ANSI X9.24 Retail Financial Services Symmetric Key Management Part 1.

Назначение: безопасная передача (с использованием предварительно распределённых симметричных ключей) и хранение любого ключевого материала, а также других чувствительных данных платёжных систем во всех компонентах платёжных систем, работающих с этими данными (центрами эмиссии, процессинговыми центрами и т.п. в соответствии с требованиями PCI PIN Security Requirements, определяющими:

- Имитовставка или цифровая подпись, должна вычисляться как за открытые поля ключевого блока, так и за зашифрованные поля, содержащие ключ или иные защищаемые данные;
- Процедура проверки целостности должна быть неотделима от процесса защиты ключей (производиться одновременно с шифрованием).

В соответствиями с требованиями PCI PIN Security, контейнер TR 31 содержит набор обязательных полей, специфичных для платёжных систем и ограничивающие операции с ключом («Key Usage» и «Mode of Use»).

Общая структура контейнера ACS X9 TR 31 – 2018



Для того чтобы скрыть истинную длину ключа, дополнение до кратности может быть выполнено и для более чем одного блока (64 или 128 бит).

Заголовок контейнера ACS X9 TR 31 – 2018

Поля обязательной части заголовка		
Байт	**Название поля**	**Описание**
0	Версия	Идентификатор версии контейнера, определяющий способ криптографической защиты и формат внутренней структуры
1-4	Длина контейнера	Общая длина контейнера, включая обязательную и опциональную часть заголовка, зашифрованные данные и поле имитовставки
5-6	**Назначение ключа (Key Usage)**	**Содержит информацию, определяющую область использование ключей в определённом сегменте платёжных систем (‘P0’ – ключ для шифрования PIN-блока, ‘B1’ – начальный DUKPT-ключ, ‘V2’ – ключ проверки криптограммы VISA PVV и др.)**
7	Алгоритм	Значение данного поля устанавливает алгоритм, в котором требуется использовать экспортируемый ключ (подробнее на слайде 15)
8	**Режим использования ключа (Mode of Use)**	**Определяет вид операции, для которой может применяться ключ (‘B’ - ключ для зашифрования и расшифрования, ‘C’ - ключ для формирования и проверки MAC, ‘D’ - ключ только для расшифрования; ‘E’ - ключ только для зашифрования, ‘G’ - ключ только для формирования MAC и др.)**
9-10	Версия ключа	Используется в порядковом номере версии ключа или иных целях, например, для обозначения номера компоненты в случае передачи компоненты ключа
11	Разрешение экспорта	Поле используется для разрешения или запрета экспорта ключа: ‘E’ — экспорт разрешён на специальном доверенным ключе, ‘N’ — экспорт запрещён.
12-13	Число опц. блоков	Определяет число дополнительных опциональных блоков, следующих за заголовком.

Внутренняя структура защищаемых данных

Описание поля	Длина	Шифрование
Длина защищаемых данных в битах (BigEndian-представление)	2B	Да
Любые защищаемые данные (например, TDES ключ вместе с битами чётности)	nB	
Дополнение до кратности 64 или 128 бит	Случайное число	

Симметричные ключи простого формата упаковываются непосредственно в байтовом представлении, для асимметричных ключей возможна упаковка в соответствии с выбранной структурой.

Упаковка пары RSA в соответствии с PKCS #1 v.2.1:

```
RSAPrivateKey ::= SEQUENCE {
version Version,
modulus INTEGER, -- n
publicExponent INTEGER, -- e
privateExponent INTEGER, -- d
prime1 INTEGER, -- p
prime2 INTEGER, -- q
exponent1 INTEGER, -- d mod (p-1)
exponent2 INTEGER, -- d mod (q-1)
coefficient INTEGER, -- (inverse of q) mod p
otherPrimeInfos OtherPrimeInfos OPTIONAL
}
```

Упаковка ключей подписи ГОСТ Р34.10-2012 в формат ASN.1 в соответствии с проектом Рекомендации «Транспортный ключевой контейнер»:

```
OneAsymmetricKey::= SEQUENCE
{
Version                    Version,
privateKeyAlgorithm        PrivateKeyAlgorithmIdentifier,
privateKey                 OCTET STRING,
Attributes                 [0] Attributes OPTIONAL,
...,
[[2:publicKey              [1]BIT STRING OPTIONAL]],
...
}
Version ::= INTEGER { v1(0), v2(1) } (v1, ..., v2)
PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
Attributes ::= SET OF Attribute
```

Назначение передаваемого ключа

Значение поля (ASCII)	Назначение ключа
‘A’	AES
‘E’	ECC
‘H’	HMAC
‘R’	RSA
‘S’	DSA (зарезервировано для будущих версий)
‘T’	Triple DES (TDEA)
Цифровое значение ‘0’-‘9’	Зарезервировано для проприетарных реализаций

Тип алгоритма HMAC указывается в специальной структуре в числе опциональных блоков

Структура позволяет расширить перечень защищаемых ключей, добавив значения, указывающие на криптографические механизмы на основе ГОСТ:

Значение поля (ASCII)	Назначение ключа
‘0’	«Магма» (ГОСТ Р 34.12-2015)
‘1’	«Кузнечик» (ГОСТ Р 34.12-2015
‘2’	HMAC (Р 50.1.113-2016, на основе ГОСТ Р 34.11-2012)
‘3’	Пара ключей подписи (ГОСТ Р 34.10-2012)

Поддержка отечественных криптоалгоритмов

Расширение значений поля VersionID:

VersionID	Описание	Заменяемый алгоритм
0x30 («0»)	Режим «Key Derivation Method» на основе блочного шифра «Магма» (64 бит)	TDES-CBC, CMAC
0x31 («1»)	Режим «Key Derivation Method» на основе блочного шифра «Кузнечик» (128 бит)	AES-CBC, AES-CMAC

Спасибо за внимание!

shkorkina@systempb.ru