УДК 004.62

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ШИФРОВАНИЯ ДАННЫХ НА УРОВНЕ СТОЛБЦОВ В SQL SERVER ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА ФИКСАЦИИ ДАННЫХ О ПРОИСШЕСТВИЯХ ДИСПЕТЧЕРАМИ СЛУЖБЫ СПАСЕНИЯ ПРИ ПРИЕМЕ ПОСТУПАЮЩИХ ВЫЗОВОВ

Афанасьева В.А., студентка гр. ИТб-181, IV курс
Ванеев О.Н., (к.т.н.) доцент кафедры ИиАПС
Кузбасский государственный технический университет
имени Т. Ф. Горбачева
г. Кемерово

В современных информационных системах автоматизации деятельности предприятия большое значение предается операциям, связанным с обработкой данных средствами вычислительной техники. Именно к таким операциям относится процесс переноса некоторой формы информации в другую среду, иначе говоря, ввод данных. Правильно организованный пользовательско-ориентированный ввод данных позволяет организовать быстрое сохранение и управление данными в системе. Процесс ввода данных играет важную роль в деятельности служб, обеспечивающих безопасность жизнедеятельности населения, в частности, службы спасения.

Единый дежурно-диспетчерский центр службы спасения города Кемерово является узлом, имеющим первостепенное значение в обеспечении связи с населением. На данный момент процесс приема и обработки входящих заявок частично автоматизирован. Уже организована возможность быстрого транзита входящего вызова к службе экстренного реагирования, введена автоматическая цифровая запись телефонных разговоров, а также внедрен операторский пульт, позволяющий автоматически определять входящий номер и местоположение звонящего.

Однако в существующей реализации бизнес-процесса не хватает технологии, позволяющей диспетчерам фиксировать сведения о поступающих вызовах в единой базе данных, что значительно усложняет процесс составления отчетов ЕДДС. В настоящее время особо сложные случаи фиксируются в системе 1С:Предприятие, но конфигурация данной системы не настроена для фиксации всех поступающих звонков, поэтому учреждение утрачивает возможность накопления и анализа данных об уже случившихся происшествиях, что в некоторых чрезвычайных ситуациях могло бы поспособствовать устранению или даже предотвращению проблемы.

В данной статье мы разберем, как можно автоматизировать процесс ввода данных о случившихся происшествиях, организовав при этом безопасное хранение конфиденциальных данных, сообщаемых заявителем.

Рассмотрим базовый бизнес процесс подразделения ЕДДС “Прием заявок о ЧС” (рисунок 1).

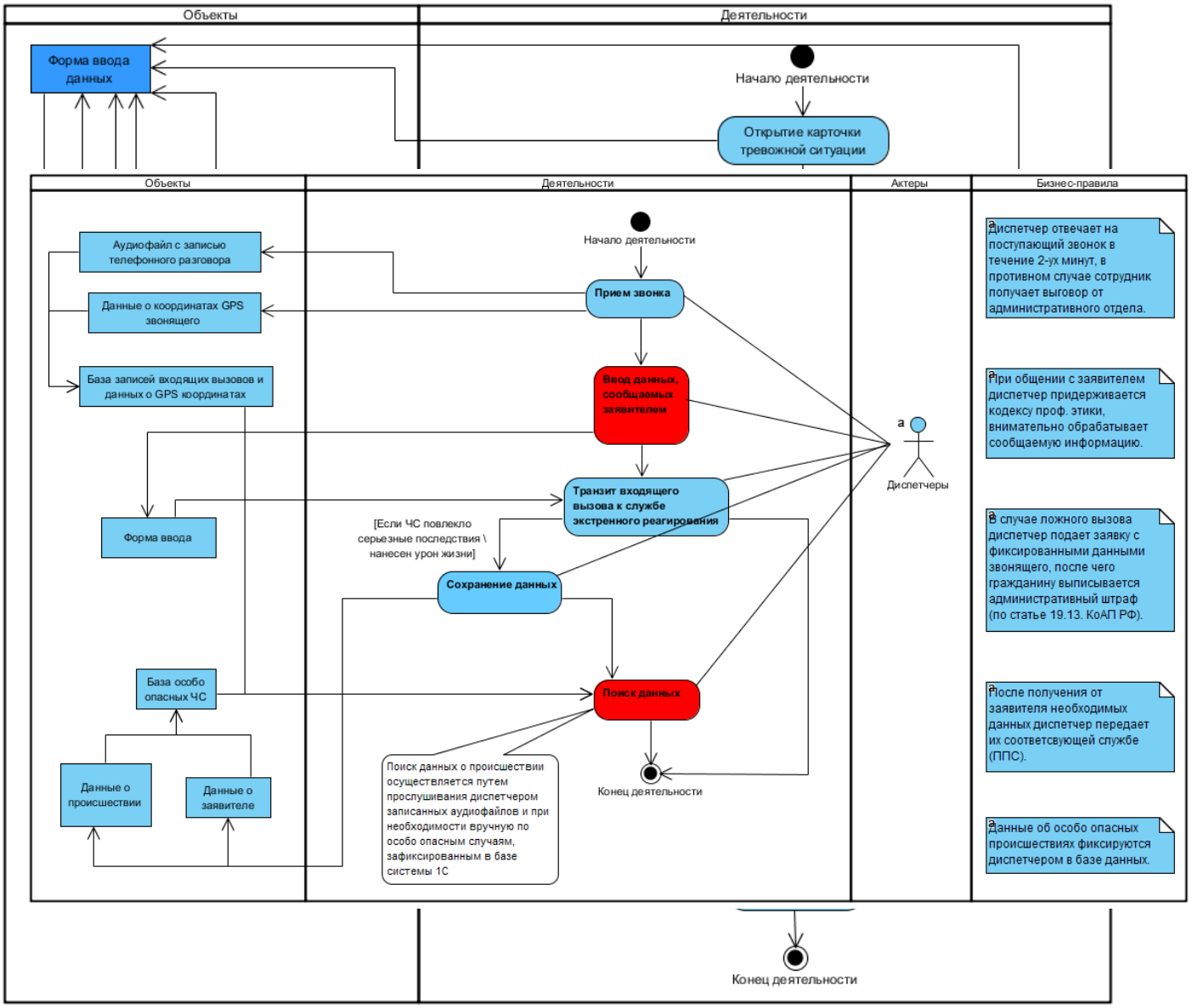
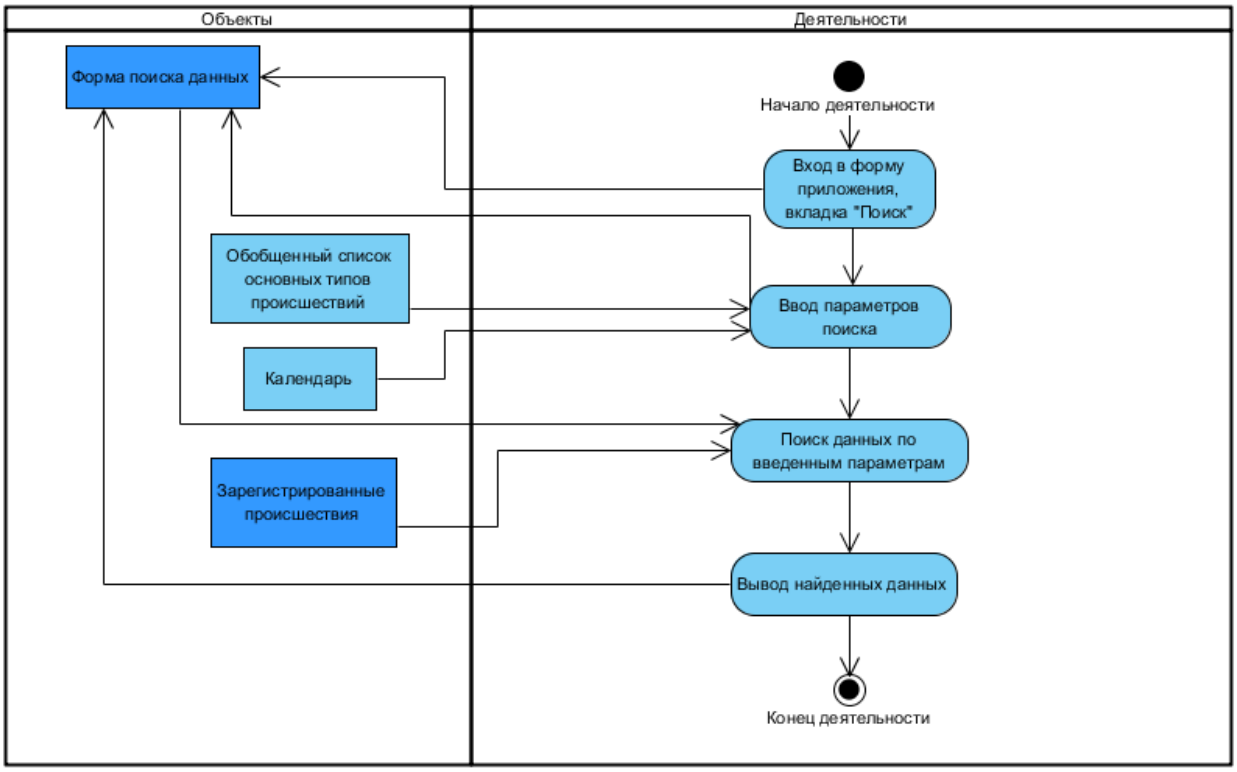


Рисунок 1 – Диаграмма деятельности процесса приема заявки о чрезвычайной ситуации.

В результате автоматизации деятельности “Ввод данных, сообщаемых заявителем” можно добиться повышения скорости фиксации диспетчером данных и возможности быстрого сохранения всех введенных данных в базе на сервере SQL. Кроме того, деятельность “Поиск данных” так же может быть автоматизирована, поскольку это позволит диспетчерам получить быстрый доступ ко всем данным о происшествиях и заявителях, хранящихся в базе службы, что значительно сократит трудозатраты при составлении отчетности. Диаграммы деятельности процессов ввода и поиска данных представлены на рисунках 2 и 3.

Рисунок 2 – Диаграмма деятельности процесса ввода данных, сообщаемых заявителем.



В данном случае целесообразнее использовать клиент-серверный принцип построения архитектуры информационной системы. В серверной компоненте будут реализовываться сценарии, связанные с доступом к данным и их основной обработкой, а в клиентской – представление данных в удобном для пользователя виде и осуществление выбора выполняемых действий. При этом клиентская часть будет реализована на языке C# при помощи интегрированной среды разработки Visual Studio Community, а серверная часть – при помощи СУБД Microsoft SQL Server Express.

Рисунок 3 – Диаграмма деятельности одного из процессов поиска данных, фиксированных в базе.

При разработке данной ИС огромное внимание необходимо уделить прежде всего проектированию серверной части, ведь система должна функционировать в муниципальном учреждении и данные, сообщаемые во время вызова гражданином, носят личный, конфиденциальный характер. Последствия от утечек такого рода данных могут оказаться серьезными и для владельцев данных, и для операторов. Для первой группы существуют многочисленные риски стать жертвой злоумышленников. Они могут пострадать от разглашения любой ин-

формации, имеющей отношение к их личности, шантажа, вмешательства в личную жизнь и т.д. Минимальным риском станет неправомерная передача сведений, например, номера телефона, каким-либо компаниям, которые начнут преследовать их обладателя рекламными объявлениями. Но даже это дает возможность возбудить дело и о неправомерной рекламе, и об утечке данных и приведет к штрафам, налагаемым на операторов, если источник утечки или спама удастся достоверно установить.

Избежать негативных последствий от утечек данных и повысить уровень безопасности можно путем использования возможностей шифрования. Шифрование - это процесс приведения данных в запутанное непонятное состояние, вследствие чего повышается уровень их безопасности.

SQL Server, как и другие распространенные коммерческие системы управления базами данных, располагает множеством вариантов шифрования, в

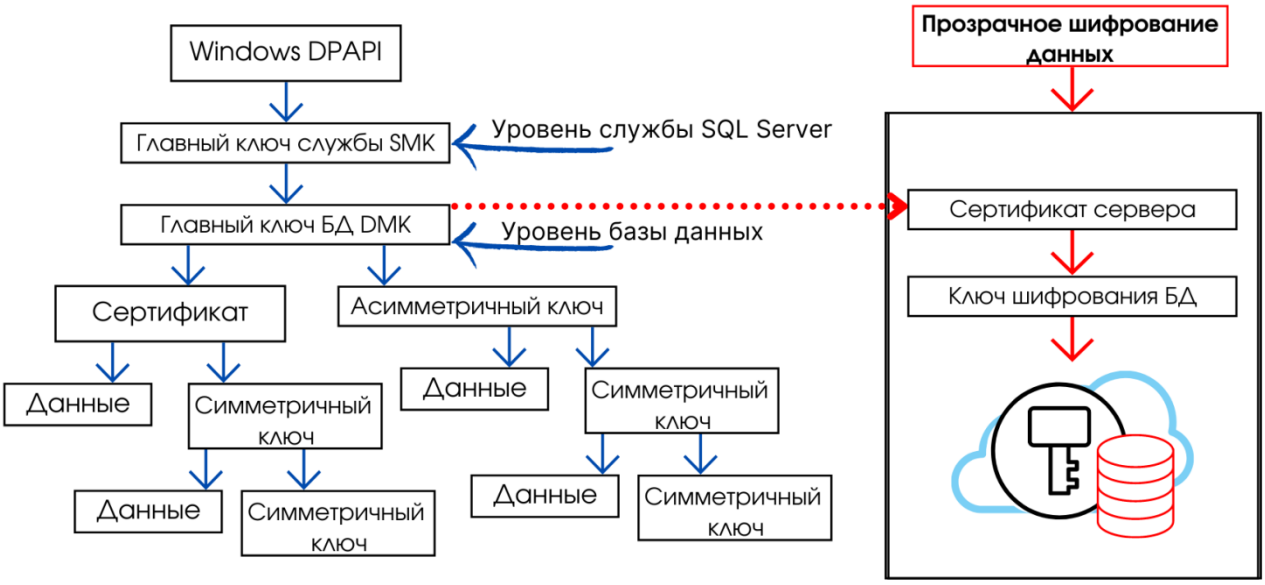
Уровень SQL SERVER	Уровень ANSI X9.17	Описание
SMK	Главный ключ	SMK - ключ верхнего уровня, используемый для шифрования DMK. SMK шифруется с применением DPAPI
DMK	Ключ шифрования ключей	DMK - симметричный ключ, используемый для шифрования симметричного ключа, асимметричного ключа и сертификата. Для каждой базы данных может быть определен только один DMK
Симметричные, асимметричные ключи и сертификаты	Ключ данных	Симметричные ключи, асимметричные ключи и сертификаты используются для шифрования данных

том числе на уровне столбцов, базы данных и файлов через Windows, а также на транспортном уровне. Эти варианты шифрования обеспечивают безопасность информации на уровне базы данных и операционной системы. Кроме того, они снижают вероятность несанкционированного раскрытия конфиденциальных сведений, даже если поражены инфраструктура или база данных SQL Server.

Модель шифрования SQL Server в основном предоставляет функции управления ключами шифрования, соответствующие стандарту ANSI X9.17. В этом стандарте определены несколько уровней ключей шифрования, использующихся для шифрования других ключей, которые в свою очередь применяются для шифрования собственно данных. Уровни ключей шифрования SQL Server и ANSI X9.17 показаны на рисунке 4.

Рисунок 4 – Уровни ключей шифрования SQL Server и ANSI X9.17.

Главный ключ службы Service master key(SMK) — ключ верхнего уровня и предок всех ключей в SQL Server. SMK — асимметричный ключ, шифруемый с



использованием Windows Data Protection API (DPAPI). SMK автоматически создается, когда шифруется какой-нибудь объект, и привязан к учетной записи службы SQL Server. SMK используется для шифрования главного ключа базы данных Database master key (DMK).

Симметричные ключи — основное средство шифрования в базе данных. Microsoft рекомендует шифровать данные только с помощью симметричных ключей. Кроме того, в SQL Server 2008 и более новых версиях есть сертификаты уровня сервера и ключи шифрования базы данных для прозрачного шифрования данных. На рисунке 5 показана иерархия ключей шифрования для SQL Server 2008 и более новых версий.

Рисунок 5 – Иерархия ключей шифрования в SQL Server 2008 и более новых версиях.

В данном случае нерационально использовать “прозрачное” шифрование данных, поскольку при использовании данного метода все данные в базе шифруются, таким образом, неконфиденциальные данные шифруются наравне с конфиденциальной информацией, что значительно расширяет вычислительные ресурсы. Таким образом, шифрование данных на уровне столбцов является более подходящим методом шифрования.

Шифрование и дешифрование данных будет реализовано при помощи языка запросов SQL с использованием ключа и сертификата.

Данный метод шифрования является частью иерархии шифрования SQL Server и поддерживает применение симметричных и асимметричных ключей, а также сертификатов.

Рисунок 6 – Типы шифрования и их функции.

Среди трех методов с точки зрения производительности наиболее эффективным будет симметричный ключ. Для демонстрации воспользуемся таблицей Applicants, в столбце Address которой хранится информация об адресе прописки заявителя.

```
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pa$$w0rd';
CREATE CERTIFICATE AddressCert
WITH SUBJECT = 'Адреса заявителей';
CREATE SYMMETRIC KEY AddressKey
WITH ALGORITHM = AES_128
ENCRYPTION BY CERTIFICATE AddressCert;
OPEN SYMMETRIC KEY AddressKey DECRYPTION BY CERTIFICATE AddressCert;
```

Для начала необходимо создать мастер-ключ библиотеки и надежный пароль для шифрования, после объявляются сертификаты и на их основе строится симметричный ключ для непосредственного шифрования данных в выбранном столбце таблицы. Кроме того, в запросе также необходимо указать алгоритм шифрования. Начиная с SQL Server 2016, все алгоритмы, кроме AES_128, AES_192 и AES_256, устарели, поэтому лучше использовать актуальную версию, например, симметричный алгоритм блочного шифрования AES_128.

Рисунок 7 – Создание ключей и сертификата.

```
GO
ALTER TABLE Applicants ADD
AddressEncrypted varbinary(8000) NULL;
UPDATE Applicants
SET AddressEncrypted = ENCRYPTBYKEY(Key_GUID('AddressKey'), Address);
```

Перед использованием ключа необходимо расшифровать симметричный

Тип шифрования	Функция шифрования	Функция дешифрования
Симметричное шифрование	**ENCRYPTBYKEY()**	**DECRYPTBYKEY()**
Асимметричное шифрование	ENCRYPTBYASYMKEY ()	DECRYPTBYASYMKEY()
Свидетельство	ENCRYPTBYCERT()	DECRYPTBYCERT()

ключ, иначе он будет недоступен для использования (последняя строка запроса). Операция открытия привязана к сеансу, а не к контексту, следовательно, ключ остается доступным до тех пор, пока сеанс не будет явно закрыт или завершен.

Далее необходимо создать новый столбец типа varbinary и вставить туда зашифрованные данные из столбца Address.

Рисунок 8 – Вызов функции шифрования адресов.

После шифрования данных необходимо удалить исходный столбец с незашифрованными адресами и обновите код, чтобы использовать новый столбец.

Рисунок 8 – Демонстрация шифрования столбца Address таблицы Applicants.

```
create procedure ReturnApplicant @IncidentID int
as
begin
OPEN SYMMETRIC KEY AddressKey DECRYPTION BY CERTIFICATE AddressCert;
SELECT SavedIncidentsSet.Info, Applicants.FIO,Applicants.Phone,
CONVERT(NVARCHAR(30), DECRYPTBYKEY(Applicants.AddressEncrypted)) AS Address, Applicants.FlatNum
From Applicants JOIN SavedIncidentsSet ON SavedIncidentsSet.Id = Applicants.IncidentID
WHERE Applicants.IncidentID = @IncidentID;
CLOSE SYMMETRIC KEY AddressKey ;
END
GO
EXEC ReturnApplicant @IncidentID = 2
```

Результаты | Сообщения

	Info	FIO	Phone	Address	FlatNum
1	Авария, столкнулись 3 машины	Иванов Артем Юрьевич	89547628945	Проспект Ленинградский 30/1	145

После шифрования данных, необходимо предусмотреть процедуры, позволяющие расшифровать их для обработки доверенными лицами (сотрудниками отдела ЕДДС).

Рисунок 10 – Реализация процедуры ReturnApplicant, расшифровывающей столбец с адресом конкретного заявителя, и демонстрация ее работы.

Шифрование надежно защищает данные и сокращает вероятность несанкционированного раскрытия конфиденциальной информации, так как без соответствующего ключа или пароля данные бесполезны.

Список литературы:

1. Шифрование и управление ключами в SQL Server. Часть 1 [Электронный ресурс] – URL: https://osp.ru/winitpro/2019/01/13054793 (дата обращения: 27.03.2022).
2. Аутентификация и шифрование данных. [Электронный ресурс] – URL: https://professorweb.ru/my/sql-server/2012/level3/3_9.php (дата обращения: 26.03.2022).

```
ALTER TABLE Applicants
DROP COLUMN Address;
select top 5* from Applicants ;
CLOSE SYMMETRIC KEY AddressKey ;
```

Результаты | Сообщения

	FIO	Phone	FlatNum	IncidentID	AddressEncrypted
1	Иванов Артем Юрьевич	89547628945	145	2	0x007256607A6F2C4C91733A465B4CE15301000000ED1432...
2	Степанова Ольга Юрьевна	89576548216	67	3	0x007256607A6F2C4C91733A465B4CE15301000000E98B5C...
3	Новоселова Марина Эдуардовна	89461257839	249	4	0x007256607A6F2C4C91733A465B4CE1530100000004B1365...

3. Основы MS SQL Server шифрования на примере симметричного шифрования. [Электронный ресурс] URL: https://dbasimple.blogspot.com/2013/07/ms-sql-server.html (дата обращения: 27.03.2022).
4. Моделирование на языке UML в среде Visual Paradigm 14. [Электронный ресурс] – URL: http://sp.cs.msu.ru/ooap/exer2017.html#exer34 (дата обращения: 24.03.2022).