A.V. Gorlatyh, P.V. Smirnov

Identity Management systems cryptographic protection technique

Key words: identity management, cryptography, informational security, shibboleth

This work proposes methods of identity management systems cryptographic protection. Now, there is no cryptographic software, which can be fully used under Russian Federation law circumstances. We propose solution, which relies on integration of open-source identity management system and certified cryptographic software.

А.В. Горлатых, П.В. Смирнов

МЕТОДИКА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ СИСТЕМ УПРАВЛЕНИЯ ИДЕНТИФИКАЦИОННЫМИ ДАННЫМИ

Введение

В результате своего развития человечество все глубже погружается в мир цифровых технологий. Рост числа электронных сервисов и приложений в информационной инфраструктуре организации приводит к тому, что становится все сложнее управлять идентификационной информацией. Вследствие отсутствие централизованного подхода в данном процессе в системах безопасности появляются уязвимости, связанные с некорректной обработкой и хранением учетных записей пользователей.

Под управлением идентификационной информацией понимается процесс автоматизации процедуры аутентификации пользователей системы и с целью контроля доступа к ресурсам системы, основанного на соответствии аутентифицированного субъекта с его правами и ограничениями.

В настоящее время активно развивается технология единого входа [1], позволяющая уменьшить количество идентификационной информации. Суть данной технологии заключается в вынесении механизма аутентификации за пределы сервиса и замене его на доверие к третьей стороне. Это позволяет стандартизировать различные сервисы и обеспечить для них единые механизмы аутентификации. В дополнение к этому пользователи смогут получать доступ ко множеству сервисов и приложений, обладая всего лишь одной парой логин-пароль. Существуют различные подходы к реализации данной технологии, один из которых нашел отражение в модели аутентификации на основе утверждений.

Выбор системы управления идентификационными данными для встраивания

В настоящее время существует большое количество как коммерческих, так и свободно распространяемых систем управления идентификационными данными (СУИД). Они отличаются как структурой, там и объемом протоколов, которые они поддерживают. Для выбора системы, используемой в методике,былвыдвинут ряд критериев, по которым происходило сравнение существующих на данный момент решений.

Выбор системы для встраивания основывался на следующих критериях:

- наличие открытого исходного кода;
- кросс-платформенность;
- поддержка популярных протоколов (WS-Federation и SAML).

Результаты сравнительного анализа представлены в табл. 1, знаком «*» отмечены системы, исходный код которых открыт для модификации.

Ни одна из представленных в табл. 1 систем не соответствует требованиям ПКЗ-2005 [2] и требованиям ФСБ к системам обработки персональных данных [3] полностью. Это значит, что такие системы не могут использоваться в государственных организациях, организациях, выполняющих государственные заказы, или организациях, обрабатывающих персональные данные. Поэтому было принято решение о встраивании сертифицированных средств криптографической защиты информации в уже существующую систему из списка.

Исходя из полученной информации, авторами была выбрана СУИД Shibboleth [4], которая легко масштабируется, обладает поддержкой различных схем аутентификации, реализует все популярные на данный момент протоколы идентификации и имеет хорошо задокументированный API, позволяющий с легкостью реализовывать дополнительные модули.

В качестве потребителя утверждений Центра идентифиации в рамках данной работы был использован программный продукт OpenAM. Причина этого заключается в том, что ShibbolethSPнаписан на язык C++ и не может использоваться на разных платформах без перекомпиляции. Использование комплекса ShibbolethIdPOpenAMSP позволяет обеспечить кросс-платформенность и простоту интеграции продуктов друг с другом. В качестве сертифицированного СКЗИ был выбран продукт КриптоПро JCP [5].

Таблица 1. Соответствие систем и поддерживаемых ими протоколов

Разработчик	**Продукты**		**Протоколы**				ГОСТ
	ЦИ	ЦП	WS-* Passive	WS-* Active	SAML	Liberty	
Atricore	JOSSO (Enterprise Edition)		+	–	+	–	–
Azigo	AzigoIdP	X	–	+	–	–	–
Bandit*	BanditIdp	X	–	+	–	–	+
CA Technologies	CA SiteMinder – Federation Manager		–	+	–	–	–
CardGears	CardGearsIdP	X	–	–	+	–	–
HP	HP OpenView IdM		+	–	+	+	–
Higgins*	SAML2 Idp 1.1	X	–	–	+	–	+
	STS IdP	X	+	+	–	–	
	X	Ext. Prot. RP Website 1.1	–	+	–	–	
IBM	IDM Tivoli Federated Identity Manager		+	+	+	+	–
Internet2 Middleware Initiative*	Shibboleth IdP	Shibboleth SP	+	–	+	–	–
JOSSO*	JOSSO (Community Edition)		+	–	+	–	–

Окончание табл. 1

Разработчик	**Продукты**		**Протоколы**				ГОСТ
	ЦИ	ЦП	WS-* Passive	WS-* Active	SAML	Liberty	
Microsoft	ADFS IdP	ADFS SP	+	–	–	–	+ (без поддержки SAML)
	ADFS 2.0 IdP	ADFS 2.0 SP	+	+	+	–	
	WIF STS	WIF RP	+	–	+	–	
Novell	Novell Access Manager		+	+	+	+	–
Oracle	Oracle IAMS (Identity Server)	Oracle IAMS (Access Server)	+	–	+	+	–
PingIdentity	Ping Federate		+	+	+	–	–
RSA Security	Federated Identity Manager		+	–	+	–	–
SimpleSAMLPhp*	SAML 2.0	SAML 2.0	+	–	+	–	–

Схема работы комплекса ShibbolethIdP-OpenAMSP

Комплекс ShibbolethIdP-OpenAMSP работает по протоколу SAML, который может быть схематично изображен в следующем виде (рис. 1).

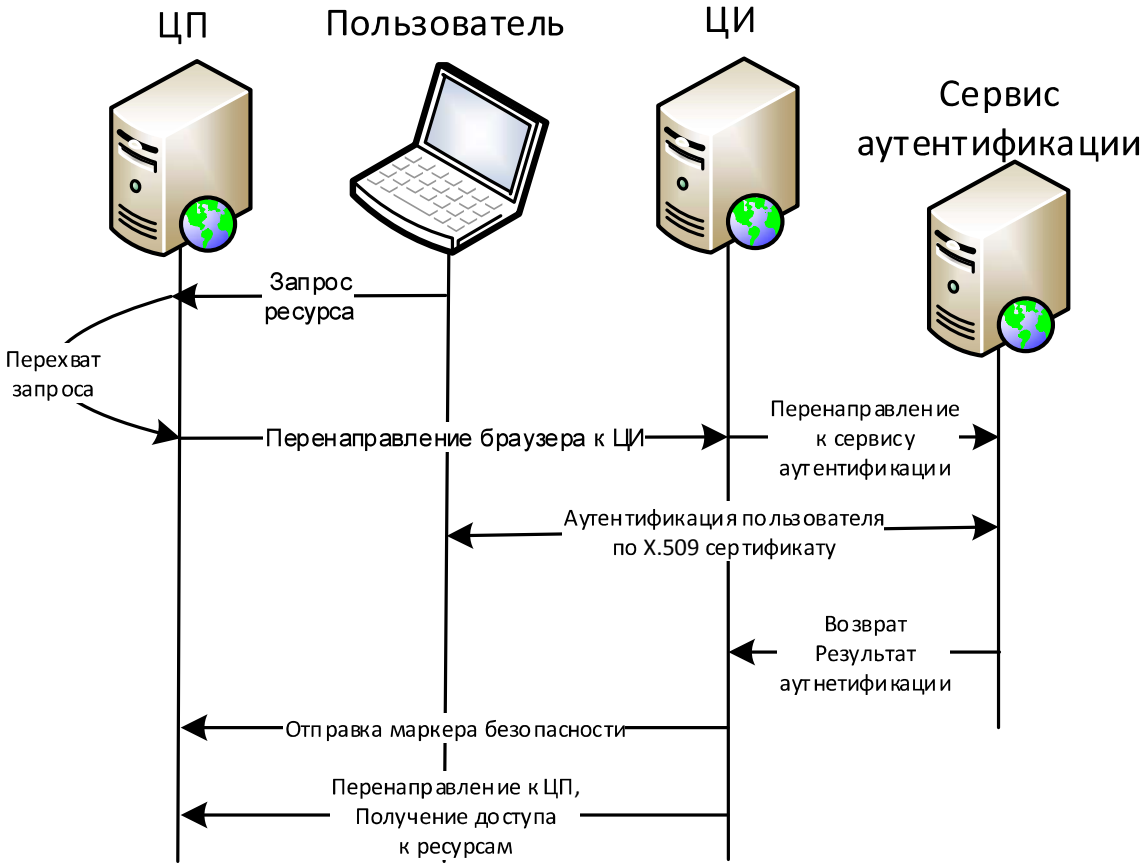

Рис. 1. Схема работы комплекса Shibbleth IdP-OpenAM SP

Перечень криптографических операций, выполняемых в процессе работы целевой схемы, имеет следующий вид:

- загрузка Центром идентификации криптографического ключа ГОСТ Р 34.10-2001 [6] для подписи маркера безопасности в сторону Целевого Приложения;
- чтение открытого ключа из метаданных Целевого Приложения;
- подпись маркера безопасности;
- генерация симметричного ключа ГОСТ 28147-89 [7];
- шифрование маркера безопасности на сгенерированном ключе ГОСТ 28147-89;
- генерация эфемерного ассиметричного ключа для выработки ключа согласования по алгоритму Диффи-Хеллмана;
- шифрование симметричного ключа на ключе согласования;
- чтение Целевым приложением закрытого ключа из своих метаданных;
- чтение открытого ключа Центра идентификации;
- чтение эфемерного ключа из маркера и выработка ключа согласования по алгоритму Диффи-Хеллмана на стороне Целевого приложения;
- расшифрование симметричного ключа шифрования ГОСТ 28147-89;
- расшифрование маркера безопасности;
- проверка подлинности цифровой подписи Центра идентификации.

Модификация механизма загрузки ключевой информации

С учетом того, что Shibboleth не хранит ключи и сертификаты в доверенном хранилище (в отличие от файлов, являющихся недоверенным хранилищем), необходимо было модифицировать механизм загрузки таким образом, чтобы данный механизм обладал возможностью загрузки информации из хранилищ, формат которых соответствовал формату хранилищ КриптоПро JCP.Модификация была выполнена таким образом, чтобы можно было в качестве типа хранилища указать все возможные типы хранилищ КриптоПро JCP. Дополнительно необходимо было модифицировать процедуры загрузки ключа и сертификата из хранилища.

Модификация механизма распознавания алгоритмов подписи

После загрузки ключевой информации и сертификатов необходимо сообщить системе Shibboletho том, как сопоставить добавленный ключевой алгоритм GOST3410 и алгоритм ЭЦП «http://www.w3.org/2001/04/xmldsig-more#gostr34102001-gostr3411».

В системе Shibboleth есть механизм динамического определения идентификатора алгоритма подписи по идентификатору алгоритма открытого ключа. Этот механизм работает следующим образом:

- система получает идентификатор алгоритма открытого ключа jceKeyAlgoID;
- далеесистемаопределяетURI (UniversalResourceIdentificator), которым в библиотеке Xmlsecописывается каждый алгоритм подписи, путем перебора спискаURI, доступных для данного ключевого алгоритма;
- после этого система вызывает метод JCEMapper.translateURItoJCEID() библиотеки Xmlsec, который получает значение идентификатора алгоритма ЭП по его URI из списка зарегистрированных в Xmlsec.

Таким образом, нам необходимо интегрировать поддержку алгоритмов ГОСТ на каждом этапе процедуры разрешения идентификатора алгоритма подписи.

Так как закрытый ключ, возвращаемый криптопровайдером КриптоПроJCP, реализует стандартный интерфейс JCAPrivateKey, который обладает методом

getAlgorithm(), нет необходимости модифицировать процедуру получения идентификатора алгоритма ключа.

Для второго пункта механизма распознавания необходимо добавитьсписок, с помощью которого будут сопоставлены URI алгоритмов цифровой подписи и идентификатор алгоритма ключа.

Модификация механизма распознавания алгоритмов шифрования

Для получения идентификатора алгоритма шифрования в системе предусмотрен метод getKeyEncryptionCredential.

Данный метод просматривает узел <SPSSODescriptor> метаданных Целевого приложения на наличие в нем ключа или сертификата. В описываемой схеме, в узле <SPSSODescriptor>содержится X.509 сертификат, из которого извлекается открытый ключ. На следующем этапе полученный ключ проверяется на соответствиенабору критериев, одним из которых является принадлежность ключа к определенному алгоритму (KeyAlgorithmCriteria). Для того, чтобы ключ ГОСТ Р 34.10-2001 был успешно загружен из сертификата, необходимо заменить KeyAlgorithmCriteria(«RSA») на KeyAlgorithmCriteria(JCP.GOST_DEGREE_NAME), где JCP.GOST_DEGREE_NAME является константой, описывающей имя алгоритма ключа.

На втором этапе осуществляется генерация симметричного ключа, алгоритм которого будет взят из конфигурации приложения. Для того, чтобы симметричным алгоритмом по умолчанию был ГОСТ 28147-89, необходимо добавить в метод populateEncryptionParams класса DefaultSecurityConfigurationBootstrap.javaоператор, который установит URIключа ГОСТ 28147-89 в качестве алгоритма ключа по умолчанию.

На следующем этапе в конфигурации Shibboleth следует указать, какой алгоритм необходимо использовать в качестве алгоритма транспортировки ключей. Это можно сделать, добавив в метод populateEncryptionParams соответствие между алгоритмами ГОСТ Р 34.10-2001 и ГОСТ 28147-89.

Методика криптографической защиты информации в системе управления идентификационными данными

После того, как был модифицирован криптографический модуль комплекса ShibbolethIdP-OpenAMSP, была разработана методика, при помощи которой была осуществлена криптографическая защита информации в комплексе.

Основными этапами методики являются:

- создание ключевых пар ГОСТ Р 34.10-2001 для шифрования и подписи;
- замена сертификатов X.509 с ключами RSA на сертификаты, содержащие ключи ГОСТ Р 34.10-2001, соответствующие закрытым ключам, созданным на предыдущем этапе;
- настройка протокола TLS для сервлет-контейнера Apache Tomcat;
- интеграция криптографических модулей в исполняемые war-файлы систем комплекса.

Создание ключевых пар осуществляется при помощи контрольной панели КриптоПро JCP. На этом этапе необходимо сгенерировать ключевые пары ГОСТ Р 34.10-2001 для подписи и шифрования, а также третью ключевую пару, которая будет использована для аутентификации сервера по протоколам SSL/TLS. В дополнение к ключам контрольная панель позволяет осуществить генерацию сертификатов, соответствующих созданным ключам, что будет использовано на следующем этапе.

Описанный в работе комплекс оперирует со следующими типами сертификатов:

- сертификат для подписи токена;
- сертификат для шифрования токена;
- сертификат для аутентификации сервера;
- корневой сертификат Удостоверяющего центра.

В дополнение к этому для корректной работы потребуется сертификат, при помощи которого пользователь сможет осуществить клиентскую аутентификацию (например, в браузере). В рамках этого этапа все сертификаты, кроме клиентского и корневого сертификата Удостоверяющего центра, должны быть помещены в хранилище КрипоПро JCP. Оставшиеся сертификаты должны быть помещены в соответствующие хранилища сертификатов операционной системы.

После того, как необходимые сертификаты будут получены, следует заменить узлы <KeyDescriptor>метаданных ЦИ и ЦП на Base64 содержимое сертификатов, соответствующих шифрованию и подписи.

Для того, чтобы ApacheTomcat мог использовать криптографические ключи ГОСТ 34.10-2001, необходимо создать в контрольной панели КриптоПро JCP, хранилище сертификатов <CertStoreName>.store, в которое будут помещены сертификаты, используемые в процедуре аутентификации клиента и сервера протокола TLS.

Настройка TLS для ApacheTomcat заключается в модификации конфигурационного файла Server.xml, содержащего информацию о коннекторах. После того, как данные изменения будут произведены, браузер сможет аутентифицировать пользователя, используя его X.509 сертификат, если он содержит ключ ГОСТ Р 34.10-2001.

Для реализации подписи и шифрования с применением алгоритмов ГОСТ необходимо выполнить модификацию криптографических модулей систем Shibbolethи OpenAM согласно руководству, описанному выше.

Заключение

В ходе работы была проведена модификация криптографического модуля комплексаShibbolethIdP-OpenAMSP, которая позволила использовать в данных продуктах криптографические преобразования, реализованные по алгоритмам ГОСТ Р 34.10-2001, ГОСТ Р 34.11-94 и ГОСТ 28147-89. Была разработана методика криптографической защиты с применением сертифицированных средств КриптоПро JCPи КриптоПроJTLS.

Предлагаемая методика позволяет использовать комплекс в соответствии с законодательством Российской Федерации в сфере криптографической защиты информации, тем самым расширяя область применения систем управления идентификационными данными. Данная методика может быть незначительно модифицирована, что позволит использовать ее в других сертифицированных средствах криптографической защиты информации с алгоритмами подписи ГОСТ 34.10-2012. В дополнение к этому, существует возможность применить данную методику к другой целевой системе управления идентификационными данными, что дополнительно потребует анализ криптографического модуля последней.

СПИСОК ЛИТЕРАТУРЫ:

1. Marc Mercuri. Beginning Information Cards And Cardspace: From Novice To Professional / Marc Mercuri – Apress – 2007 - 428 p.
2. Приказ ФСБ РФ от 9 февраля 2005 г. N 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)» – Зарег. в Минюсте РФ 3 марта 2005 г., рег. N 6382
3. ФЗ-152 «О персональных данных» от 14 июля 2006 [Электронный ресурс]. URL: http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=LAW;n=144649 (дата обращения: 04.04.2014)
4. Shibboleth Identity Provider [Электронныйресурс] Режим доступа к ресурсу: https://shibboleth.net/products/identity-provider.html(дата обращения: 04.04.2014)
5. КриптоПро JCP [Электронный ресурс]. URL: http://www.crypto-pro.ru/products/csp/jcp (дата обращения: 04.04.2014)
6. ГОСТ Р34.10-2001 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – Введ. 2002–07–01. – М.: Изд-во стандартов, 2001. – 16 с.
7. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Введ. 1990–01–01. – М.: Изд-во стандартов, 2001. – 28 с.

REFERENCES:

1. Marc Mercuri. Beginning Information Cards And Cardspace: From Novice To Professional / Marc Mercuri – Apress – 2007 - 428 p.
2. PrikazFSBRFot 9 fevralya 2005 g. N 66 «ObutverzdeniiPolozeniyaorazrabotke, proizvodstve, realizacii iekspluatacii shifrovalnih (kriptographicheskih) sredstv zashiti informatsii (Polozanie PKZ-2005)» – Zareg. v MinusteRF 3 marta 2005 g., reg. N 6382
3. FZ-152 «O personalnih dannih» ot 14 iulya 2006. – Zareg. v MinusteRF27 iunya 2006g.
4. Shibboleth Identity Provider. URL:https://shibboleth.net/products/identity-provider.html
5. КриптоПро JSP.URL:https://www.crypto-pro.ru/products/csp/jcp;
6. GOST R 34.10-2001 Informacionnaya tehnologiya. Kriptographicheskaya zashita informacii. Procesiformirovaniyaiproverkielektronnoi cifrovoi podpisi. – Vved. 2002–07–01. – M.: Izdatelstvostandartov, 2001. – 16 s.
7. GOST 28147-89 Sistemi obrabotki informacii. Zashita kriptographicheskaya. Vved. 1990–01–01. – M.: Izdatelstvostandartov, 2001. – 28 c.