**НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ**

**БАКЪЛАВЪРСКИ ФАКУЛТЕТ**

**ДЕПАРТАМЕНТ "ИНФОРМАТИКА"**

**ПРОГРАМА** **ИНФОРМАТИКА**

**ДИПЛОМНА РАБОТА**

**ТЕМА: AS2 (Applicability Statement 2) сървър и клиент за трансфер на данни**

**НА СТУДЕНТА: Илиян Делчев Додеков ФАК. № 50129**

**НАУЧЕН РЪКОВОДИТЕЛ: Илиян Занкински**

**София**

**2015**

Съдържание

[Уводна част 3](#_Toc432079420)

[I. Резюме 3](#_Toc432079421)

[II. Мотивация 4](#_Toc432079422)

[III. Актуалност 5](#_Toc432079423)

[Обзорна част 6](#_Toc432079424)

[I. История 6](#_Toc432079425)

[II. Цялостен процес 6](#_Toc432079426)

[Използвана литература 7](#_Toc432079427)

# Уводна част

## Резюме

AS2 (Applicability Statement 2) е спецификация за това как да се трансферира информация сигурно и гарантирано в Интернет. Сигурността се постига с помощта на дигитални сертификати и криптиране.

AS2 е базиран на HTTP и S/MIME. Това е вторият AS протокол разработен и ползва същите конвенции за подписване, криптиране и MDN, както оригиналният AS1 от 90-те години. С други думи:

* Файловете са „закачени“ (attachment) към стандартизирано S/MIME съобщение (наричано още AS2 съобщение).
* AS2 съобщенията винаги се изпращат по HTTP или HTTPS протокол (Secure Sockets Layer не е задължителен) и обикновено с помощта на POST метод.
* Съобщенията могат да бъдат компресирани, но не е задължително.
* Съобщенията могат да бъдат подписани с помощта на дигитален сертификат, но не е задължително.
* Съобщенията могат да бъдат криптирани с помощта на дигитален сертификат, но не е задължително.
* Съобщенията могат да поискат Message Disposition Notification (MDN), за потвърждение, че трансферът е успешен. Искането на MDN потвърждение не е задължително. MDN представлява съобщение, което следва нормите за трансфер на данни според AS2 протокола, но има специфично съдържание.
* Ако MDN потвърждение е поискано:
  + Когато дадено съобщение е получено, успешно декриптирано и подписа е верифициран, „успешен“ MDN ще бъде изпратен на изпращача на съобщението. MDN обикновено се подписват, но не се криптират.
  + Когато MDN съобщението бъде получено, декриптирано и подписа е верифициран, оригиналният изпращач ще знае, че трансферът е успешен (така наречената особеност – non-repudiation).
  + Ако има проблеми с трансфера може да бъде пратен „неуспешен“ MDN. Според протоколната спецификация обаче, липсата на MDN също означава „неуспех“.

Обикновено при AS2 трансфер има изискване двете страни, които обменят информация, предварително да си разменили набор от сертификати и идентификатори (т. нар. „имена на партньорите“). Имената на партньорите могат да бъдат всякаква валидна фраза.

AS2 спецификацията е описана подробно в RFC 4130 (Request for comment 4130). Дипломната работа ще имплементира стандарта до степен, в която следните функционалности са реализирани (както са описани в RFC 4130):

* HTTP транспорт
* S/MIME съдържание на съобщенията
* Криптиране с помощта на X509 дигитален сертификат
* Подписване с помощта на X509 дигитален сертификат
* Синхронен или асинхронен MDN

За осъществяването на успешен AS2 трансфер, софтуерът, който ще бъде изработен трябва едновременно да играе ролите на клиент и сървър (тоест изпращач и получател). За да се постигне това, ще се имплементира Java базирано решение.

За да се изпрати файл, разработеният продукт ще следи преконфигурирана директория за един или повече файлове и ще ги изпраща на даден получател посредством AS2. От друга страна, при получаване на съобщение, цялата информация ще бъде извлечена под формата на един или повече файлове и ще бъде записана в преконфигурирана директория.

Продуктът ще има възможност за проследяване на това как изглежда AS2 съобщението, когато се праща по мрежата. За да може да се демонстрира функционалността, продуктът ще бъде инсталиран два пъти, за да може двете инсталации да комуникират помежду си.

## Мотивация

Преди да взема решението за темата на дипломната работа, направих обстойно проучване какъв AS2 софтуер се предлага на пазара. Поради високата сигурност, сложност и изисквания на протокола, повечето решения таргетират ентърпрайз пазара, в резултат на което са доста скъпо платени. От друга страна решенията, които са безплатни и/или с отворен код са доста неразвити и неактуални. Повечето безплатни решения ползват доста стари технологии, в резултат на което имат неактуални имплементации. Също така се спират до реализацията на протоколната имплементация и нямат фокус на лесната работа с тях и имплементирането на функционалности, които да ги направят наистина използваеми в реалния живот.

Целта на тази дипломна работа е да се разработи AS2 комуникационен софтуер, който да покрива протоколната спецификация. Идеята е впоследствие да се подобри решението, така че да бъде възможно ползването му в реална клиентска среда. В същото време приложението ще бъде с отворен код.

## Актуалност

Въпреки че спецификацията е финализирана в RFC 4130 през Юли 2005 година, тя все още е актуална и всъщност е една от най-ползваните в световен мащаб. Това се дължи отново на факта, че протоколът се ползва предимно от ентърпрайз пазара за сигурен трансфер на данни – предимно EDI документи. Както знаем, промените в ентърпрайз пазара се внедряват доста бавно и като цяло, за да започне да се ползва дадена технология, тя трябва първо да се докаже. AS2 е добър пример за доказана технология.

От 2013 година се стандартизира и AS4 спецификацията, която до голяма степен спазва същите принципи за сигурност, но ползва SOAP протокол за транспорт и сигурност. AS4 като стандарт тепърва ще се доказва и ще навлиза в пазара. Неговата идея не е да замести AS2, а по-скоро дава възможност да бъде ползван от организации, които са искат да ползват или са инвестирали в web services(<http://www.drummondgroup.com/index.php/component/content/article/127-b2b/b2b-products/b2b-faqs/243-as4-faq> ).

# Обзорна част

## История

AS2 (Applicability Statement 2) спецификацията е предложена от Drummond Group и стандартизирана през 2005 година от IETF (Internet Engineering Task Force). Спецификацията дефинира как се транспортират Electronic Data Interchange (EDI) съобщения сигурно през Интернет с HTTP протокол. Въпреки, че протокола е направен с умисъл за транспорт на EDI документи, на практика може да се ползва за всякакъв тип файлове.

AS2 надгражда AS1 спецификацията, която ползва S/MIME за транспорт. Като цяло, в момента съществуват 4 applicability statement протокола – AS1, AS2, AS3, AS4. Колективно се наричат **ASx**. Разликите в тях могат да бъдат сведени до типа на протокола, който се използва за транспорт:

|  |  |
| --- | --- |
| Протокол | Транспорт |
| AS1 | **S/MIME** |
| AS2 | **HTTP** |
| AS3 | **FTP** |
| AS4 | **SOAP** |

## Цялостен процес

Важно е преди да започне да се говори за спецификите в AS2 да се опише цялостния процес за трансфер на файл от край до край.

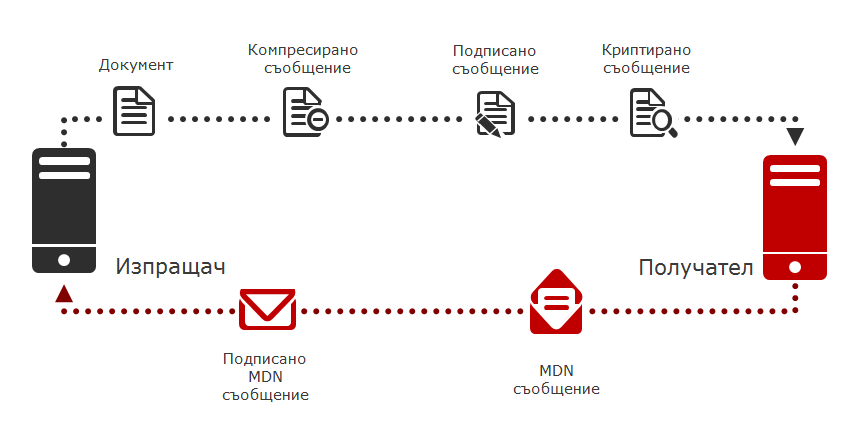
За да може да говорим за AS2 комуникация между 2-ма партньори, е нужно всеки от тях да има инсталиран и конфигуриран AS2 сървър. Освен това за успешна комуникация е нужно партньорите да си разменят набор от X509 сертификати и идентификатори (повече за това в следващите глави на този документ).

За изпращането на файл от партньор “Алфа” до партньор “Бета” се следват следните стъпки:

1. Алфа закача файла към стандартизирано MIME съобщение (наричано още AS2 съобщение).
2. Алфа може да компресира съобщението, но не е задължително.
3. Алфа може да подпише съобщението, но не е задължително.
4. Алфа може да криптира съобщението, но не е задължително.
5. Алфа може да отбележи в съобщението, че иска да получи потвърждение при успешен трансфер. Съобщенията могат да поискат Message Disposition Notification (MDN), за потвърждение, че трансферът е успешен. Искането на MDN потвърждение не е задължително. MDN представлява съобщение, което следва нормите за трансфер на данни според AS2 протокола, но има специфично съдържание.
6. След като съобщението успешно се пакетира, Алфа го изпраща до Бета по HTTP с POST заявка.
7. Бета получава съобщението и започва процес по разпакетиране.
8. Бета декриптира съобщението, ако то е криптирано.
9. Бета верифицира дигиталния подпис, ако съобщението е било подписано.
10. Бета декомпресира съобщението, ако то е било компресирано.
11. Бета проверява дали в съобщението е отбелязано, че трябва да се изпрати MDN за успешното получаване на съобщението.
12. Ако MDN потвърждение е поискано, Бета генерира MDN съобщение, със специално съдържание, съдържащо параметри от оригиналното съобщение.
13. Ако е поискано, Бета подписва MDN съобщението. При MDN съобщенията няма компресиране и криптиране.
14. Бета изпраща MDN съобщението на Алфа.
15. Алфа получава MDN съобщението и започва процес по разпакетиране
16. Алфа верифицира дигиталния подпис, ако MDN съобщението е било подписано.
17. Алфа проверява съдържанието на MDN съобщението и за кое оригинално съобщение се отнася потвърждението.
18. Ако съдържанието на MDN съобщението съвпада, трансферът приключва.

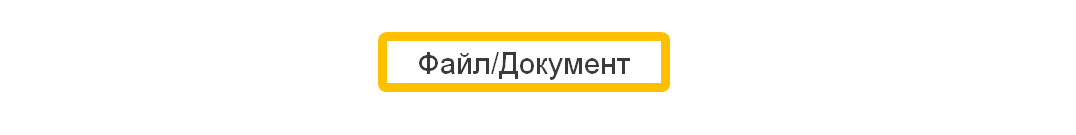
Всяка от тези стъпки е обяснена по-подробно в следващите глави на този документ.

По-горният процес може да бъде обобщен в следващата графика:

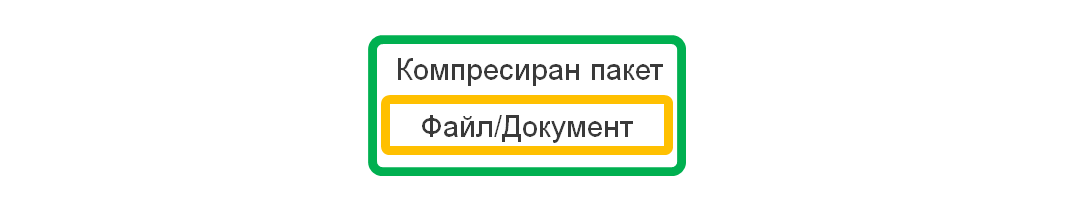


За по-лесно илюстриране на различните трансформации при пакетирането на едно AS2 съобщение, то преминава през следните етапи, като никои от етапите не е задължителен:

1. Обикновен файл.



1. Компресиран файл.



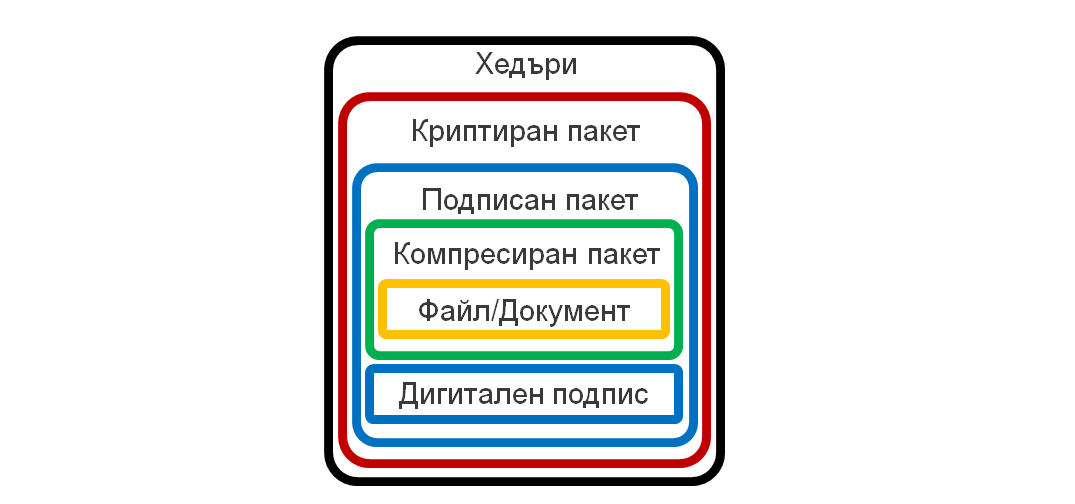
1. Подписано съобщение



1. Криптирано съобщение



1. AS2 съобщение



Важно е да се отбележи, че последователността от трансформации винаги се извършва в този ред – компресиране -> подписване -> криптиране. Всяко от тези стъпки може да бъде пропусната в зависимост от конфигурацията.

Същото правило важи и при получаване на съобщение, но в обратен ред – декриптиране -> верифициране на подписа -> декомпресиране.

## Използвани технологии

За реализацията на приложението е ползван Java 8, като програмен език. От уикипедия (<https://bg.wikipedia.org/wiki/Java>):

Java или Джава е обектно-ориентиран език за програмиране. Кодът, написан на Java не се компилира до машинен код за определен процесор, а до специфичен за езика код, наречен байт код. Поради това за изпълнението на програма, написана на Java е необходима т. нар. Виртуална машина (на английски: Java Virtual Machine).

Подобна реализация има своите предимства и недостатъци. Сред главните предимства са:

* лесната поносимост между различните платформи (софтуерни или хардуерни) - веднъж написана и компилирана, една Java-програма може да бъде стартирана на компютри независимо от архитектурата или от операционната им система. За целта е необходимо само да се инсталира виртуална машина за съответната платформа;
* допълнителните действия, извършвани от виртуалната машина като освобождаване на паметта от обекти(класове) които не се използват(Garbage collector), проверка за размерността на масивите;
* възможността за контрол на правата на потребителя на ниво виртуална машина
* първоначално заделяна на heap, част от паметта резервирана за джава
* висока степен на сигурност поради факта, че програмистите не работят директно с паметта и др.

Главен недостатък е необходимостта от допълнителни ресурси (под формата на процесорно време и памет) за изпълнението на самата виртуална машина.

Съществуват и компилатори, които превеждат байт кода до машинен код. Недостатъка при използването им е, че тогава програмата не може да се стартира на различни платформи.

Избраният език за имплементацията е Java, за да може приложението да е платформено незвисимо и да може да се изпълнява на всякаква операционна система.

Използвани са също следните външни Java библиотеки за подпомагането на реализацията на някои от компонентите на приложението:

|  |  |  |
| --- | --- | --- |
| Библиотека | Версия | Описание |
| Bouncy Castle Cryptography API | 1.52 | Колекция от програмни интерфейси, ползвани в криптографията. |
| Bouncy Castle Mail | 1.52 | Програмен интерфейс за работа с S/MIME и email съобщения. |
| Apache HTTP Client | 4.5 | Библиотека позволяваща генериране на клиентски HTTP заявки. |
| Java Mail API | 1.5.4 | Програмен интерфейс за работа с S/MIME и email съобщения. |
| Jetty Server | 9.3 | Библиотека позволяваща работата с HTTP уеб сървър. |
| Apache Log4j | 1.2.17 | Библиотека за работа с лог файлове. |
| Java Servlet API | 3.1 | Библиотека позволяваща имплементирането на сървлети за даден уеб сървър. |

# Използвана литература

<https://en.wikipedia.org/wiki/AS2>

<https://en.wikipedia.org/wiki/AS4>

<http://www.ld.com/as2-part-1-what-is-it/>

<http://www.ld.com/as2-part-2-best-practices/>

<http://www.drummondgroup.com/index.php/component/content/article/127-b2b/b2b-products/b2b-faqs/243-as4-faq>

[http://www.ietf.org/rfc/rfc4130.txt](https://www.ietf.org/rfc/rfc4130.txt)

<http://www.ietf.org/rfc/rfc2616.txt>

<http://www.ietf.org/rfc/rfc2045.txt>

<http://www.ietf.org/rfc/rfc2633.txt>

http://www.ietf.org/rfc/rfc1847.txt

<http://www.ietf.org/rfc/rfc2376.txt>

http://www.ietf.org/rfc/rfc2298.txt

<http://www.drummondgroup.com/b2b-certified-products/b2b-standards/as2>

<https://moveitsupport.ipswitch.com/support/micentral/help/MICAS1AS2AS3Overview.htm>

<https://bg.wikipedia.org/wiki/Java>

<https://www.bouncycastle.org/java.html>

<https://java.net/projects/javamail/pages/Home>

<http://hc.apache.org/httpcomponents-client-ga/>

<http://www.eclipse.org/jetty/>

<https://logging.apache.org/log4j/1.2/>